creative computing

Nov-Dec 1976
vol 2, no 6

Austria Sch 32.00
Belgium BF 70.00
Canada $1.50
Denmark Kr 11.00
Finland Mk 7.00
France F 9.00
Germany DM 4.50
Great Britain £ 1.00
Greece Dr 65.00
Holland Dfl 5.00
Italy L 1500
Norway Kr 10.00
Portugal Esc 55.00
Spain Ptas 120.00
Sweden Kr 8.00
Switzerland SF 4.50
USA $1.50

the magazine of recreational and educational computing

$1.50

Frederik Pohl: A Day in the Life of Able Charlie

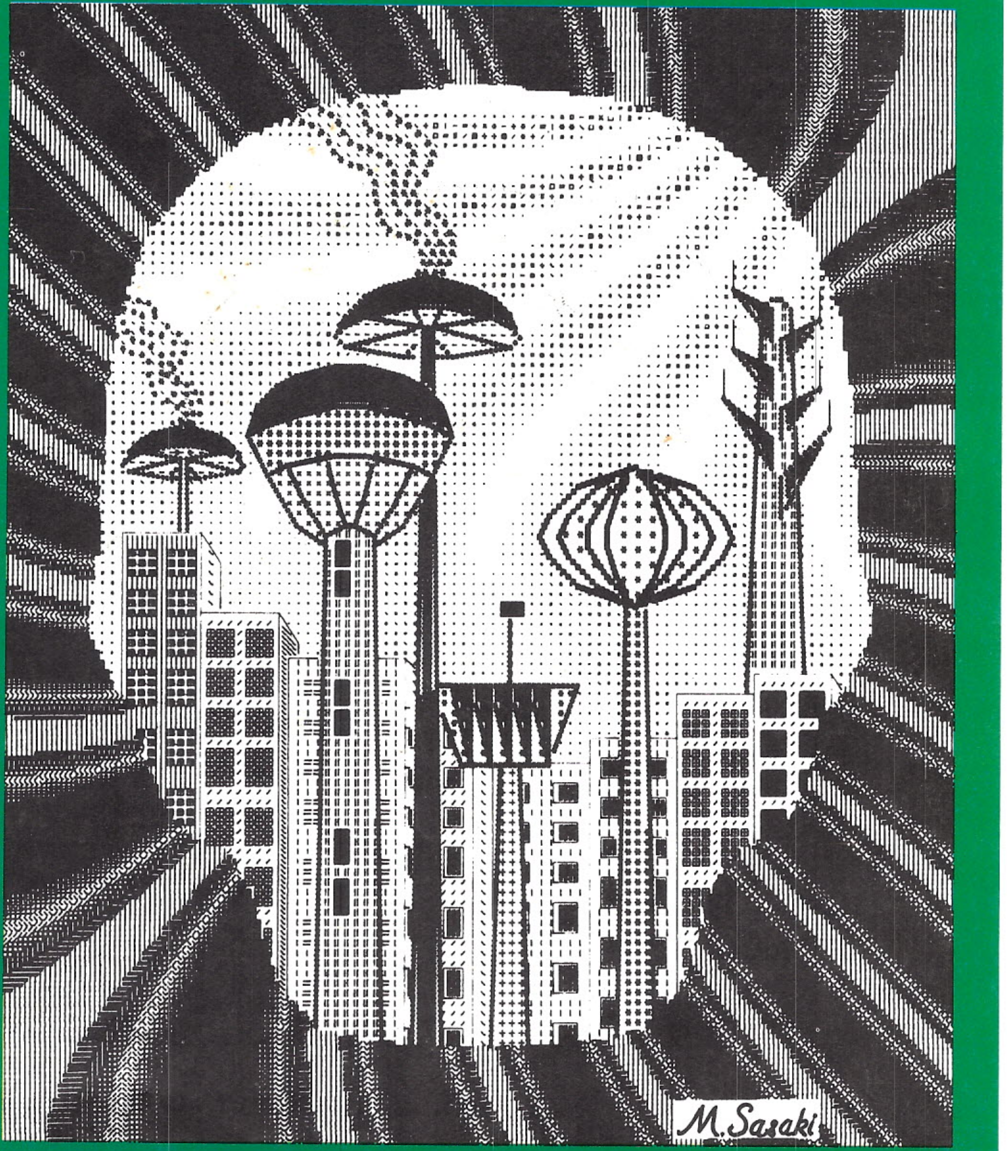
Beyond BASIC

Games: Poster, Two-to-Ten, LEM

Computer Sorting Techniques

Computer Literacy Quiz

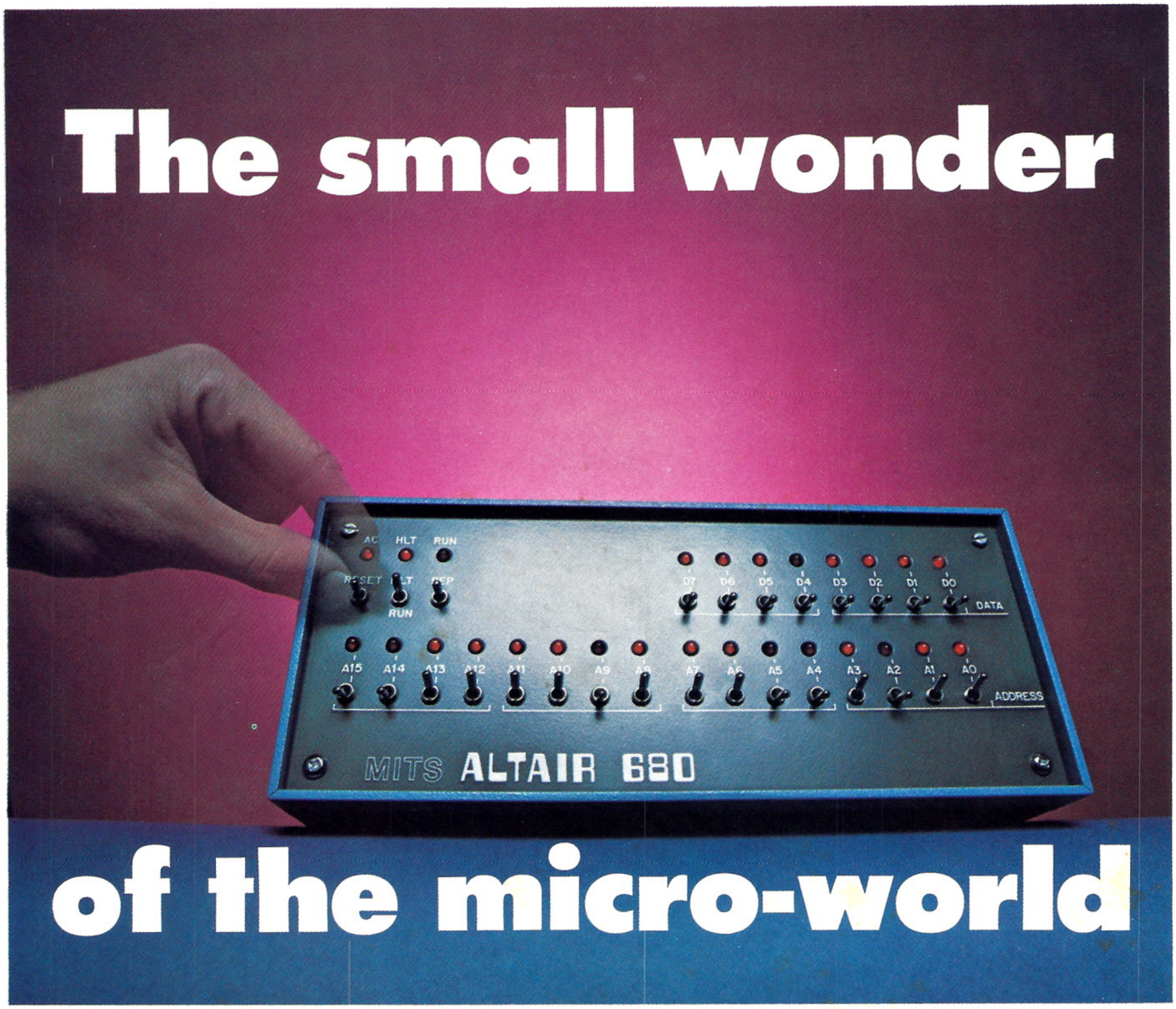
Computers and Beauty



Computer Stores

Equipment Reviews: Tektronix 4051, HP 25, Odyssey Game

Luehrmann: Should the Computer Teach the Student or Vice Versa?



Measuring just 11″ wide x 11″ deep x 5″ high, and weighing a mere 7 pounds, the Altair ™ 680b is a complete, general-purpose computer.

The secret to this revolutionary, small computer is its CPU board. This double-sided board fits along the bottom of the Altair case and plugs directly into the front panel board. It contains the new 6800 microprocessor, 1,024 bytes of RAM memory, a 256 byte PROM monitor, provisions for 768 bytes of additional PROM or ROM, and a single Interface port with a Motorola ACIA serial interface adapter which can be configured either RS-232 or TTY. A five level Baudot interface option is also available.

The Altair 680b can be programmed from front panel switches, or it can be interfaced to a video display terminal, or teletypewriter. Three additional circuit boards can be plugged inside the Altair 680b for further memory and interface expansion. The first of these boards is a 16K static RAM memory board.

Software already developed includes Altair 680 BASIC with all the features of the 8K BASIC previously developed for the Altair 8800. These include Boolean operators, the ability to read or write a byte from any I/O port or memory location, multiple statements per line, and the ability to interrupt program execution and then continue after the examination of variable values. This software takes only 6.8K bytes of memory space and a copy is **included free with the purchase of the Altair 680 16K memory board.**

Other software includes a resident two pass assembler.(also free with 16K board) The Altair 680b is also compatible with Motorola 6800 software.

The Altair 680b is ideal for hobbyists who want a powerful computer system at an economic price. Altair 680b owners qualify for membership in the Altair Users Group, and like other Altair owners, they receive a complimentary subscription to **Computer Notes** and complete factory support.

NOTE: Altair is a trademark of MITS, Inc.

PRICES:

Altair 680b kit with complete, easy-to-understand assembly manual, operator's manual, and programming manual	$466
Assembled Altair 680b	$625
Altair 680b Turnkey model kit	$395
Expander Card 680MB (required to expand 680)	$ 24
Altair 680BSM 16K static RAM board kit with 680 BASIC	$685
Altair 680 BASIC when purchased separately	$200
Baudot option	$ 42

MAIL THIS COUPON TODAY

□ Enclosed is a check for $__________
□ BankAmericard #__________ or Master Charge #__________
□ Altair 680b □ Kit □ Assembled □ Other (specify)
enclose $8 for postage and handling (Continental U.S. Others subject to quote)
□ Please send free information package.

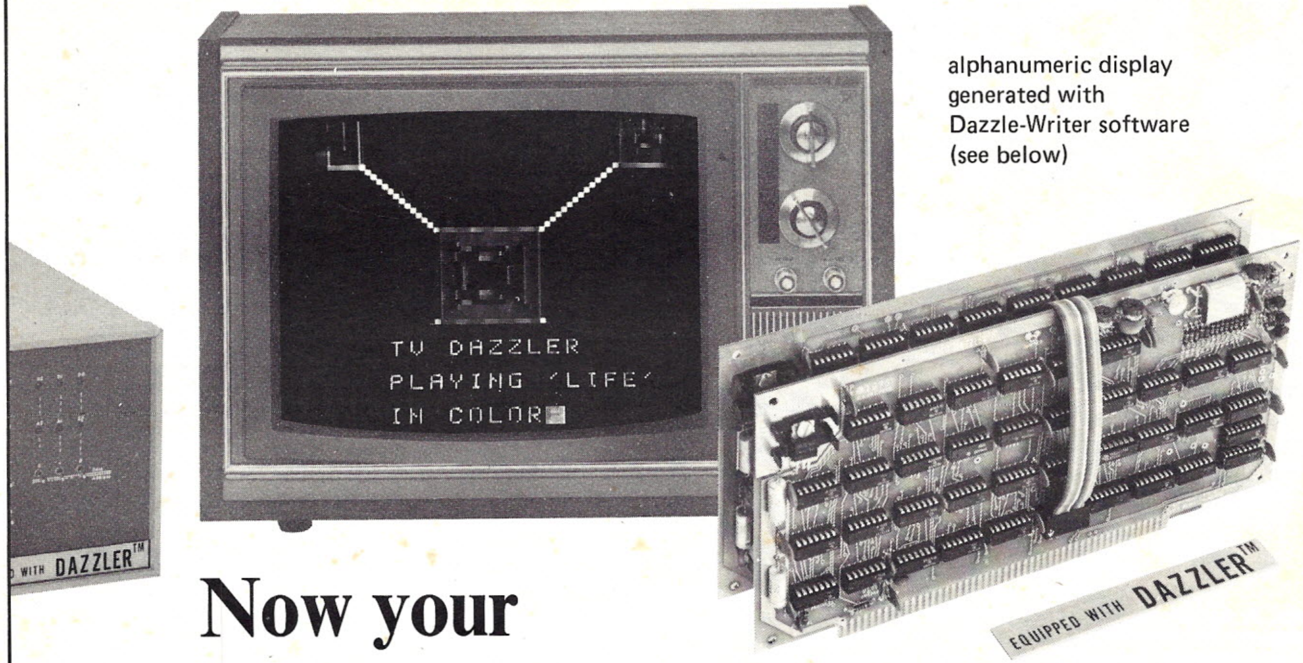
NAME____________________

ADDRESS____________________

CITY__________ STATE & ZIP__________

mits
2450 Alamo SE/Albuquerque, NM 87106, 505-243-7821

Price, specifications subject to change. Please allow up to 60 days for delivery.



```
TV DAZZLER
PLAYING 'LIFE'
IN COLOR
```

alphanumeric display
generated with
Dazzle-Writer software
(see below)

Now your color TV can be your computer display terminal

New capabilities, too

Cromemco's new computer/tv interface circuit lets you have a full-color computer display terminal for little more than a black-and-white terminal.

The Cromemco interface also lets you do vastly more with your color terminal than you can do with ordinary black-and-whites.

We call our interface the TV Dazzler®. It consists of two circuit boards that plug directly into your Altair 8800 or IMSAI 8080 computer.

Alphanumerics plus action, and graphics

The Dazzler® maps your computer memory content onto your color tv screen in full color.

That doesn't mean just that you see alphanumerics in color. You can display *any* information in memory. And do so in color.

LIFE in color

You can display computer games or animated shows (rocket ships). What's more, you can display business or technical graphics — multi-colored charts, graphs, histograms, educational material — all from computer memory. Even light shows. Not even the biggest computer manufacturers offer all this in color.

Needs only 2K memory

Technically, the Dazzler® scans your computer memory using direct-memory access (DMA). It formats each memory bit into a point on the tv screen to give a 128 x 128-element picture. Only a 2K-byte computer memory is required (only 512 bytes for a 32 x 32 picture). The quality of the pictures is evident in the photos.

The Dazzler® output is a video signal that goes directly to the tv video amp or to the antenna terminal through an inexpensive commercially-available device.

Inexpensive — and so much better

You can see from the list below that the Dazzler® is little if any more in price than an ordinary b/w interface or tv typewriter. But it does so much more.

Order now

By mail or at your computer store

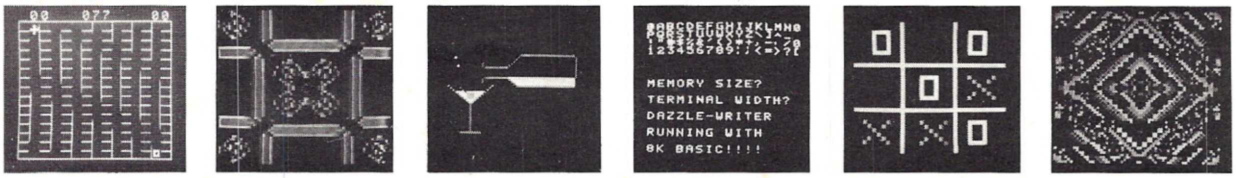
If you're into computers (or want to be), if you want to invent these beautiful displays or games, or to plot colorful material inexpensively at home or in business, the Dazzler® is for you.

Not only is it reasonable, but it's sold at computer stores from coast to coast.

Or order directly by mail on your bank card.

TV DAZZLER® (complete kit)	$215
TV DAZZLER® (fully assembled and tested)	$350
SOFTWARE (punched paper tape with documentation)	
LIFE in full color	$15
KALEIDOSCOPE in full color	$15
DAZZLE-WRITER (for alphanumeric displays in color)	$15
DAZZLE-MATION (for computer-generated animated displays	$15
TIC-TAC-TOE (you play the computer)	$15

Shipped prepaid if fully paid with order.
California users add 6% sales tax.
Mastercharge and BankAmericard accepted with signed order.
Delivery: from stock for immediate shipment.



```
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ\]^_
!"#$%&'()*+,-./0
123456789:;<=>?[

MEMORY SIZE?
TERMINAL WIDTH?
DAZZLE-WRITER
RUNNING WITH
8K BASIC!!!!
```

CHASE • LIFE • DAZZLEMATION • DAZZLE-WRITER • TIC-TAC-TOE • KALEIDOSCOPE
SOME SEQUENCES FROM CROMEMCO SOFTWARE

Cromemco
Specialists in computers and peripherals
2432 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

SCELBI SOFTWARE

SCELBAL
A HIGHER LEVEL LANGUAGE FOR 8008/8080 SYSTEMS
SCELBI COMPUTER CONSULTING INC.

Introducing SCELBAL, the new microcomputer language that's simpler than machine language.

A complete language for "8008"/"8080" systems including source listings, routines, flow charts and more!

SCELBAL. SCientific **EL**ementary **BA**sic **L**anguage for "8008"/"8080" systems. A complete, illustrated program book. Routines. Techniques. Source Listings. Flow Charts. And more. Took several years to develop. Now yours for many years to come. First time that intimate details of higher level language has been offered for such a low price. Only $49! You get **5 Commands:** SCR, LIST, RUN, SAVE, LOAD. **14 Statements:** REM, LET, IF . . . THEN, GOTO, FOR with STEP, END, INPUT, PRINT, NEXT, GOSUB, RETURN and optional DIM. **7 Functions:** INT, SGN, ABS, SQR, RND, CHR, TAB. And, it runs in 8K and more. Here's all the data needed to customize a high level language for your system . . . at a fraction of the cost!

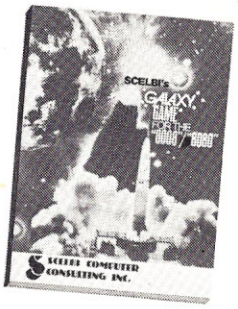
Order your copy today! Get started advancing your system! $49 ppd.

SCELBI's FIRST BOOK of COMPUTER GAMES for the "8008"/"8080"
SPACE CAPTURE
HEXPAWN
HANGMAN
SCELBI COMPUTER CONSULTING INC.

Here's SCELBI's First Book of Computer Games for the 8008/8080. Action-packed. And fun.

Try to beat the computer at its own game. Here's the first complete machine language computer manual for computer games to include source listings, flow charts, routines and more. **Space Capture** — You against the computer using "search and destroy" strategy to shoot down roaming alien spaceships in outerspace. **Hexpawn** — a mini-chess game that lets the computer make mistakes . . . but only once. **Hangman** — an updated version of the great kid game. Computer selects words at random from long, expandable list. Try to beat it in 8 moves or less. Illustrated. Fun extras to put your computer to challenging, competitive, fun use. **Order yours today! 14^{95} ppd.**

SCELBI's new microcomputer Game Book of Outer Space . . . GALAXY


SCELBI's GALAXY GAME FOR THE "8008"/"8080"
SCELBI COMPUTER CONSULTING INC.

Captain your own crusading starship against the logic of your "8008" or "8080". You must destroy a random number of alien ships without running out of stardates, out of fuel, out of ammunition or out of the galaxy. Plan your mission. How much fuel is used for your warp factor? Don't run into roaming stars that can damage your ship. **Suddenly! Condition Red! Alien in sight!** How big is he? Fire a phasor or torpedo! He's damaged or destroyed. But, you've used valuable fuel. That's just the beginning. A new game every time. Complete program book in machine language for 4K memory, including source listings, flow charts, routines, etc. Great intergalactic adventure and fun.

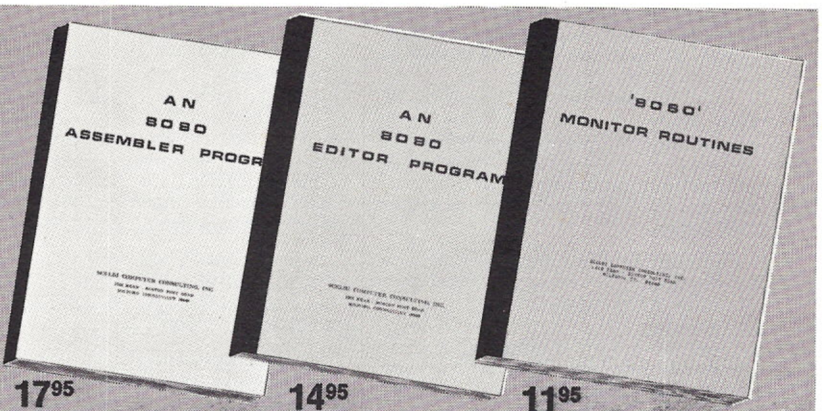
Order "GALAXY" today! 14^{95} ppd.

"8080" SOFTWARE MANUALS

Three new basic, complete "MUST" manuals! **"8080" Assembler Program** operates easily in 4K bytes of RAM (including symbol table). Unique feature: How to accept mnemonics related to "8008" based machines on "8080". Includes all major routines, pertinent flow charts, highly commented assembled listing and more. **$17.95.**

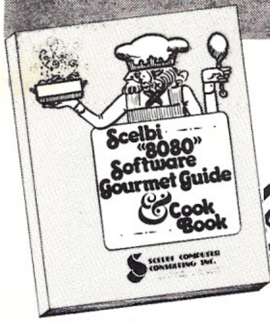
"8080" Editor Program identical to "8008" Editor, with machine codes for "8080". **$14.95.**

"8080" Monitor Routines same functions as "8008" specifically developed to utilize expanded capabilities of "8080". **$11.95.**



17^{95} ppd. **14^{95} ppd.** **11^{95} ppd.**

SAVE $5.35! BUY ALL 3 FOR ONLY $39.50!


Scelbi "8080" Software Gourmet Guide & Cook Book
SCELBI COMPUTER CONSULTING INC.

Now you can cook-up hot programs for your own "8080"

Gobble up practical "how to" facts, including the description of "8080" instruction set. How to manipulate the "8080" stack. Flow charts and source listings. General purpose routines for multiple precision operation. Programming time delays for real time applications. Random number generators. And many other basic programming function comestibles.

The "Cook Book" even includes a completely assembled floating point arithmetic program . . . plus input/output processing for your basic I/O programming through interrupt processing. Code and numeric conversion routines. Real time programming. Search and sort routines. And many, many more finger-lickin' goodies. Order your copy of **Scelbi's "8080" Software Gourmet Guide & Cook Book** today. Only $9.95 ppd.

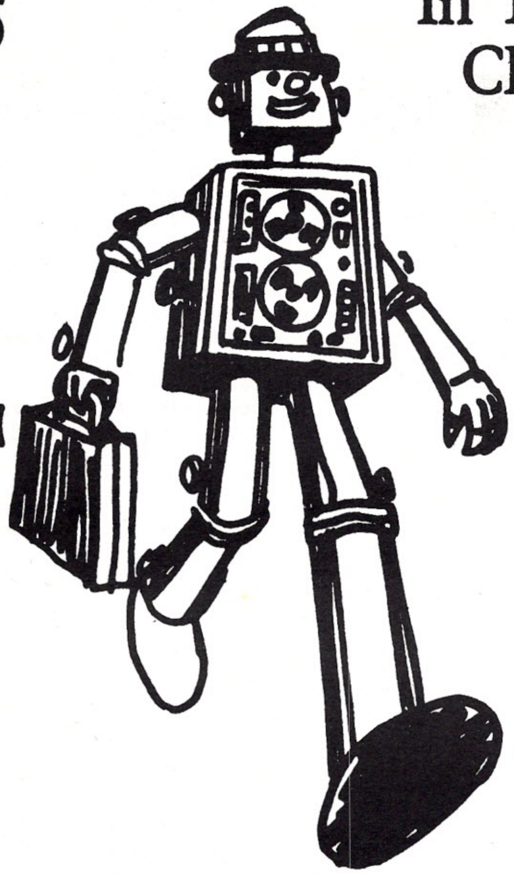
Bon appetite!

Prices shown for North American customers. Master Charge, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice.

SCELBI COMPUTER CONSULTING INC.

1322 Rear Boston Post Rd., Milford, CT 06460
Telephone: 203/874-1573

creative computing



In This Issue of CREATIVE COMPUTING

ARTICLES

FICTION & FOOLISHNESS

REVIEWS AND RESOURCES

THINGS TO DO, COMPUTER GAMES

DEPARTMENTS

Publisher
David H. Ahl

Editor
Burchenal Green

Associate Editors
John Lees, University of Missouri-Rolla
Trish Todd, Brown University
Steve North, County College of Morris
Walter Koetke, Lexington High School

Reviews Editors
Peter Kugel, Boston College
Stephen B. Gray, Amateur Computer Society

Careers Editor
Eleanor Corr, Keystone Jr. College

Higher Education Editors
A. Kent Morton, Dartmouth College
Merl Miller, Matrix Publishers

Humor Editor
Jack Ludwig

Technology Editors
Thomas R. Kibler, Georgia State University
Alan Salisbury, U.S. Army

Business Manager
Carol R. Tick

Nov-Dec 1976
Volume 2, Number 6
Consecutive Issue No. 12 *Under 1,000,000 Circulation*

Creative Computing magazine is published bi-monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (Editorial office: 20 Lynnfield Drive, Morristown, NJ 07960)

Subscriptions: Institutional 1-year $15, 3-year $40; Individual 1-year $8, 3-year $21; Foreign 1-year $10, 3-year $27. Subscription orders, change of address, Form 3579 to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Second class postage paid at Morristown, New Jersey and at additional mailing offices.

MEMBER
COSMEP
COMMITTEE OF SMALL MAGAZINE EDITORS AND PUBLISHERS

THE COVER

The cover illustration, "A City," is by M. Sasaki and T. Sasaki. Original size is 23 x 19.5cm². The computer used was a Facom 230-75. A description of the technique with additional illustrations can be found in the article "Computers and Beauty" on pp. 48-51.

Notices, etc.

WINNER

Dr. Thomas Dwyer, author of "The Art of Education: Blueprint for a Renaissance," in the last issue of *Creative Computing*, won the MOS Technology Inc.'s drawing at the Personal Computing Conference at Atlantic City. His prize was a free 4K Memory Board for a Kim-1.

Moral: Submit your next exciting article to *Creative Computing*.

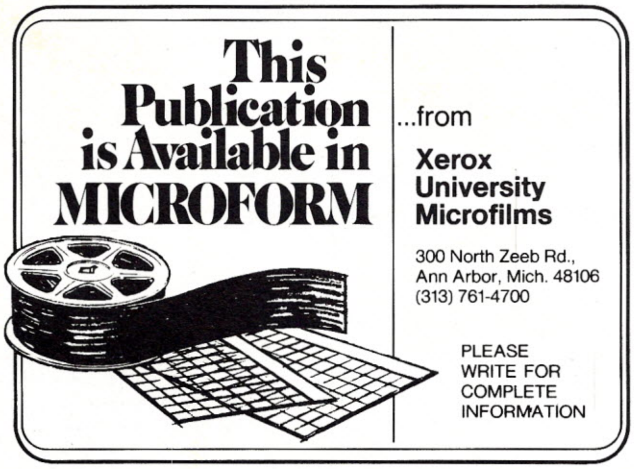
SEMINAR

E & L Instruments will hold seminars nationwide that will be hands on sessions working with computers. One instructor will be John Titus, co-author of the Bugbook series. The $80.00 fee for two days is returned in $30.00 worth of Bugbooks and a $50.00 discount on any E & L microcomputer hardware. Contact: Richard Vuilleques, E & L Instruments, Inc., 61 First Street, Derby, CN 06418

OUR FACE IS RED

"How We Spent Our Summer Vacation," Sep/Oct 1976, pp 22-25 was not, in fact, written by a group of anonymous computer bums as our lack of a credit line might imply. Author was John Lees, assisted by Richard Freeman, Dennis Keats, and Susan Culwell.

Looks like one catalogue entry, Sep/Oct 1976, pg 16, got all the mistakes: name of book should be "The Calculus With Analytic Geometry Handbook" (not Textbook), author is Jason R. Taylor (not Joseph), and price is $2.95 (not unknown).



1977 National Computer Conference
June 13-16 Dallas

SPECIAL CALL FOR PAPERS IN PERSONAL COMPUTING

The 1977 National Computer Conference will feature several events for personal computing enthusiasts including the Personal Computing Fair, exhibits of personal computing equipment by manufacturers, seminars and social events in addition to paper presentation.

Two days of Personal Computing paper and panel presentations are being planned. Papers and panels in any subject of interest to personal computing enthusiasts are sought including

- personal computer software,
- hardware designs and trends for personal computing,
- innovative applications of personal computing systems,
- the influence of the personal computing movement on the computer industry and computer science education,
- standards for personal computing products,
- predictions of trends in personal computing.

The '77 NCC will be the year's largest gathering of data processing users and computer professionals. Approximately 30,000 people are expected to gather for the conference program of over 100 sessions plus the year's largest display of computer hardware, software, systems, and services featuring over 250 exhibitors.

1977 NCC Paper Guidelines

Previously unpublished papers are solicited. Papers submitted for consideration must be in final form with all figures and tables, ready for typesetting. All papers will be refereed. Refereed and approved papers will be sent immediately to the printer, with no opportunity for author changes. The Conference Proceedings Editors reserve the right to edit all papers prior to publication or to request that the authors change them to meet AFIPS publication requirements.

The material submitted should include:

1) Six copies of the paper. The paper should be a maximum of 5,000 words. The submitted paper should be the final version—cleanly typed, double spaced on one side of the paper, ready for typesetting. Each page should be numbered and have the principal author's name on it. Submission of a paper implies guarantee by the author that all necessary approvals and clearances have been obtained.

2) Six copies of a page containing a 150 word abstract, the Computing Reviews Classification and four to six keywords descriptive of the content of the paper.

3) Three copies of a short biography of the presentor to be used in conference publicity and for introductions by the session chairman.

Deadline for all submissions is December 1, 1976. Authors will be notified before March 1, 1977 regarding the acceptance of their papers.

Please send all submissions to the '77 NCC Program Chairman:

Dr. Robert R. Korfhage
Dept. of Computer Science
Southern Methodist University
Dallas, TX 75275
214/692-3082

101 Basic Computer Games

David H. Ahl. An anthology of games and simulations—from Acey-Deucey to Yahtzee, all in the BASIC language. Contains a complete listing, sample run, plus a descriptive write-up of each game. Our most popular book! Large format, 248 pp. $7.50 [6C]

What to Do After You Hit Return

Another collection of games and simulations—all in BASIC—including number guessing games, word games, hide-and-seek games, pattern games, board games, business and social science simulations and science fiction games. Large format. 158 pp. $6.95 [8A]

Fun & Games with the Computer

Ted Sage. "This book is designed as a text for a one-semester course in computer programming using the BASIC language. The programs used as illustrations and exercises are games rather than mathematical algorithms, in order to make the book appealing and accessible to more students. The text is well written, with many excellent sample programs. Highly recommended." —*The Mathematics Teacher* 351 pp. $5.95 [8B]

Game Playing With the Computer, 2nd Ed.

Donald Spencer. Over 70 games, puzzles, and mathematical recreations for the computer. Over 25 games in BASIC and FORTRAN are included complete with descriptions, flowcharts, and output. Also includes a fascinating account of the history of game-playing machines, right up to today's computer war games. Lots of "how-to" information for applying mathematical concepts to writing your own games. 320 pp. 1976 $14.95 [8S]

BYTE Magazine

If you are considering a personal computing system now or later, BYTE provides a wealth of information on how to get started at an affordable price. Covers theory of computers, practical applications, and of course, lots of how-to build it. Monthly. 1-Year sub'n $12.00 [2A], 3-Years $30.00 [2B]

Games & Puzzles Magazine

The only magazine in the world devoted to games and puzzles of every kind — mathematical, problematical, crosswords, chess, gomoko, checkers, backgammon, wargames, card games, board games, reviews, competitions, and more. Monthly. 1-Year sub'n $12.00 [3A]

Games With The Pocket Calculator

Sivasailam Thiagarajan and Harold Stolovitch. A big step beyond tricks and puzzles with the hand calculator, the two dozen games of chance and strategy in this clever new book involve two or more players in conflict and competition. A single inexpensive four-banger is all you need to play. Large format. 50 pp. $2.00 [8H]

Games, Tricks and Puzzles For A Hand Calculator

Wally Judd. This book is a necessity for anyone who owns or intends to buy a hand calculator, from the most sophisticated (the HP65, for example) to the basic "four banger." 110 pp. $2.95 [8D]

Want to get up to speed quickly?
Then try some of these best sellers from the Creative Computing Book Service.

Creative Computing Magazine

So you've got your own computer. Now what? *Creative Computing* is chock full of answers — new computer games with complete listings every issue, TV color graphics, simulations, educational programs,. how to catalog your LPs on computer, etc. Also computer stories by Asimov, Pohl, and others; loads of challenging problems and puzzles; in-depth equipment reports on kits, terminals, and calculators; reviews of programming and hobbyist books; outrageous cartoons and much more. *Creative Computing* is *the* software and applications magazine of personal and educational computing. Bi-monthly.
1-year sub'n $8.00 [1A], 3- years $21.00 [1B], sample copy $1.50 [1C]

The Best of Creative Computing — Vol. 1

David Ahl, ed. Staggering diversity of articles and fiction (Isaac Asimov, etc.), computer games (18 new ones with complete listings), vivid graphics, 15 pages of "foolishness," and comprehensive reviews of over 100 books. The book consists of material which originally appeared in the first 6 issues of *Creative Computing* (1975), all of which are now out of print. 324 pp. $8.95 [6A]

Computer Lib/ Dream Machine

Ted Nelson. This book is devoted to the premise that everybody should understand computers. In a blithe manner the author covers interactive systems, terminals, computer languages, data structures, binary patterns, computer architecture, mini-computers, big computers, microprocessors, simulation, military uses of computers, computer companies, and much, much more. Whole earth catalog style and size. A doozy! 127 pp. $7.00 [8P]

Computer Power and Human Reason

Joseph Weizenbaum. In this major new book, a distinguished computer scientist sounds the warning against the dangerous tendency to view computers and humans as merely two different kinds of "thinking machines." Weizenbaum explains exactly how the computer works and how it is being wrongly substituted for human choices. 300 pp. $9.95 [8R]

Artist and Computer

Ruth Leavitt, ed. Presents personal statements of 35 internationally-known computer artists coupled with over 160 plates in full color and black & white. Covers video art, optical phenomena, mathematical structures, sculpture, weaving, and more. 132 pp. $4.95 [6D] Cloth cover $10.95 [6E]

Computer Science: A First Course (2nd Ed.)

Forsythe, Keenan, Organick, and Stenberg. A new, improved edition of this comprehensive survey of the basic components of computer science. There has been an updating of important areas such as Programming, Structured Programming, Problem Solving, and other Computer Science Concepts. The quantity of exercises and problems has been increased. 876 pp.$16.95 [7D]

Mr. Spock Poster

Dramatic, large (17" x 23") computer image of Mr. Spock on heavy poster stock. Uses two levels of overprinting. Comes in strong mailing tube. $1.50[5B]

Problems For Computer Solution

Gruenberger & Jaffray. A collection of 92 problems in engineering, business, social science and mathematics. The problems are presented in depth and cover a wide range of difficulty. Oriented to Fortran but good for any language. A classic. 401 pp. $8.95 [7A]

A Guided Tour of Computer Programming In Basic

Tom Dwyer and Michael Kaufman. "This is a fine book, mainly for young people, but of value for everyone, full of detail, many examples (including programs for hotel and airline reservations systems, and payroll), with much thought having been given to the use of graphics in teaching. This is the best of the introductory texts on BASIC."—*Creative Computing* Large format. 156 pp. $4.40 [8L]

BASIC Programming 2nd Ed

Kemeny and Kurtz. "A simple gradual introduction to computer programming and time-sharing systems. The best text on BASIC on almost all counts. Rating: A+"—*Creative Computing*. 150 pp. $8.50 [7E]

Build Your Own Working Robot

David Heiserman. Complete plans, schematics and logic circuits for building a robot. Not a project for novices, this robot is a sophisticated experiment in cybernetics. You build him in phases and watch his capabilities increase and his personality develop. Phase I is leash led, Phase II has a basic brain, while Phase III responds and makes decisions. 238 pp. 1976 $5.95 [9M]

Computers and Society

R. Hamming. Provides a framework for thinking about and drawing conclusions about how machines should be used in our society. Presents, in a non-technical way, the principles of computer operations, programming and use. 288 pp. 1972 $7.95 [8T]

Problem Solving: The Computer Approach

LaFave, Milbrandt, and Garth. Describes the process of thinking through the steps needed to solve a problem, flowcharting the steps, coding in a computer language, development of appropriate test data, and manual checking. 176 pp. 1973 $10.40 [8U]

Problem Solving With The Computer

Ted Sage. This text is designed to be used in a one-semester course in computer programming. It teaches BASIC in the context of the traditional high school mathematics curriculum. There are 40 carefully graded problems dealing with many of the more familiar topics of algebra and geometry. Probably the most widely adopted computer text. 244 pp. $5.95 [8J]

A Simplified Guide to Fortran Programming

Daniel McCracken. A thorough first text in Fortran. Covers all basic statements and quickly gets into case studies ranging from simple (printing columns) to challenging (craps games simulation). 278 pp. $8.75 [7F]

Understanding Solid State Electronics

An excellent tutorial introduction to transistor and diode circuitry. Used at the TI Learning Center, this book was written for the person who needs to understand electronics but can't devote years to the study. 242 pp. $2.95 [9A]

Microprocessors

A collection of articles from *Electronics* magazine. The book is in three parts: device technology; designing with microprocessors; and applications. 160 pp. 1975 $13.50 [9J]·

Microprocessors: Technology, Architecture and Applications

Daniel R. McGlynn. This introduction to the microprocessor defines and describes the related computer structures and electronic semi-conductor processes. Treats both hardware and software, giving an overview of commercially available microprocessors, and helps the user to determine the best one for him/her. 240 pp. $12.00 [7C]

The Art of Computer Programming

Donald Knuth. The purpose of this series is to provide a unified, readable, and theoretically sound summary of the present knowledge concerning computer programming techniques, together with their historical development. For the sake of clarity, many carefully checked computer procedures are expressed both in formal and informal language. A classic series. Vol. 1: Fundamental Algorithms, 634 pp. $20.95 [7R]. Vol. 2: Seminumerical Algorithms, 624 pp. $20.95 [7S]. Vol. 3: Sorting and Searching, 722 pp. $20.95 [7T].

ALGOL by Problems

B. Meek. A set of programming exercises, both abstract and concrete, to give the reader a thorough working knowledge of ALGOL. Good companion to vendor's language manual. 168 pp. 1972 $8.95 [8V]

Computer Algorithms and Flowcharting

G. Silver and J. Silver. A straightforward approach to analyzing problems and structuring solutions suitable for the computer. Branching, counters, loops, and other important concepts are presented in easily-grasped modular units in the text. 176 pp. 1975 $6.95[8W]

Creative Computing Catalogue

Zany 12-page tabloid newspaper/catalog lists books, magazines, art prints, and T-Shirts. A conversation piece even if you don't order anything. Free. [5A]

Use Order Form On Inside Back Cover.

Input/Output

Humanistic Computer Uses

Dear Editor:

I am working on a survey of the humanistic use of computers and would like to get in touch with other people who are interested in this field. I want to learn about what can, and is, being done to make the use of computers as humanized as possible. More specifically, I want to know:

—What are the potentially humane applications of computers?
—What are the important ingredients in humanized computer systems?
—What is currently being done in the field of humane computing, both in terms of theory and practice?
—Which individuals and groups are actively doing this work?
—What are their particular goals and objectives?
—What are the results so far?
—What are the major factors determining their successes and failures?

Do computers have a role in the humanizing of our society? Can we use them to facilitate the creative expression of our individuality and the healthy fulfillment of our potential? Can the use of computers help us to communicate meaningfully with each other, so as to promote mutual caring, understanding and respect? If computer systems can indeed contribute to furthering humanistic ideals, what would such systems look like in principle and how well do existing and planned systems measure up to these aims in practice?

Or perhaps computers, by their nature, are inappropriate for such a role in people's lives? If so, why, and what can be done to ensure that computer systems are as humanized as possible? Just what are the limits and limiting factors in the humanistic use of computers?

I would like to hear from anyone who can contribute to answering such questions. They should write to me describing their thoughts, fantasies, plans and experiences related to the humane use of computers.

Andrew Clement
789 West 18th Avenue
Vancouver, B.C.
Canada V5Z 1W1

Humanistic Computer Jobs?

Dear Editor:

This may sound strange but the whole idea of humanizing computers is what has been pushing me along for my degree. I'm looking around now for a job paying good money, doing just that. If you have any ideas let me know.

Mark S. Mayes
Boston University

Editor's note. If anyone has any ideas for Mark, knows of any jobs, or shares his concern, let us hear from you.

Strange Phenomenon

Dear Editor:

I recently came across a strange phenomenon which may be of some interest to you. Through a head injury which is caused by gassing it is possible to transmit sound and pictures over a wireless wave from head to head over long distances. Individual heads are turned into what could indeed be called small radio-television transmitters and receivers.

As far as I can make out the gassing enables an electrical field which is normally enclosed within the body on a circuit of its own to be tuned into. This forms a circuit over which signals can be transmitted.

In transmitting there isn't any frequency fade. The pitch of the signals seems to be several tones higher and several tones lower than ordinary sound. The sound isn't audible to the ear in the usual way. It is heard within the head or out from it according to the way that it is transmitted. It is possible that the ears are affected in such a way that they 'hear' outside the normal range of sound. I think that the sound could be thought of as being in a quite new and unresearched tonal range.

Several people combine together to tune into the head. Broadcasts between heads are almost continuous. It has occurred to me that it might be possible to tune into such broadcasts on specially made metre bands in ultra high frequencies. The fact that pictures can be transmitted as well as sound suggests that broadcasts take place on very high frequencies.

No research is being done here in Australia on this problem. I understand that this kind of communication falls into the category of Cybernetics, a relatively new field about which very little is known. Both the Russians and Japanese are carrying out studies where it is concerned.

To my mind the thing is a security risk in that in times of war transmissions could be made other than by orthodox means. I realize that all this will sound incredible to you but I hope that you will be able to give it some serious thought.

If the frequencies on which transmissions are taking place could be determined it might be possible to design a piece of electronic equipment like a small transistorized transmitter that could block signals being sent into individual heads. Enormous numbers of people seem to be involved in it here. Is any research being done in this field in America?

E. Rowlands
Sidney, NSW, Australia

Health Care Applications

Dear Editor:

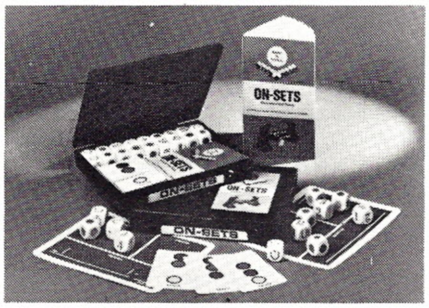
I would be very interested in corresponding with anyone in the health professions or with an interest in health care delivery data collection and recording about programs such as [the one I wrote for an IBM 5100 to collect clinical data].

Our intention is to begin in a very basic (no pun intended) manner and slowly graduate to more sophisticated programs on interacting health care problems. It is not our intention to write

Games for Thinkers

from WFF 'N PROOF publishers

THE EMPHASIS IS NOT ON WHAT TO THINK, BUT HOW TO THINK!



ON-SETS

2 to 4 players

$10.00*

The Game of Set Theory, By Layman E. Allen, Peter Kugel, (M.I.T.), Martin Owens (Mitre Corp.) The game of set theory. This 30-game package can instruct students in kindergarten through high school in basics of set theory as well as extend a challenge to any adult exploring "new math." Encourages players to enjoy mathematics while learning concepts of union, intersection, logical differences, complement, identity, inclusion, null and universal sets.

CONFIGURATIONS: Number Puzzles and Patterns for All Ages, by Harold L. Dorwart (Professor of Mathematics, Trinity College). A series of ingenious geometric and mathematical problems that will delight and challenge those who enjoy careful reasoning. Suitable for Jr. and Sr. high school levels. If you prefer solitaire, this one is for you.
1 Player $6.75*

WFF 'N PROOF

2 to 4 players

$13.00*

The Game of Modern Logic, By Layman E. Allen, (Professor of Law and Research Social Scientist, University of Michigan). The original game of symbolic logic. Twenty-one-game kit that starts with speed games that challenge intelligent adults. The kit provides entertainment and practice in abstract thinking relevant for philosophy, mathematics, English, and computer programming for Jr. and Sr. high school.

QUERIES 'N THEORIES

For teen-agers and adults

2 to 4 players

$13.00*

The Game of Science & Language. (A Simulation of Scientific Method and Generative Grammars), By Layman E. Allen and Joan Ross (University of Michigan) and Peter Kugel (Massachusetts Institute of Technology). Through inductive reasoning, players learn scientific method of inquiry and gain skill in organizing, analyzing, and synthesizing data while engaged in an intriguing game of linguistics.

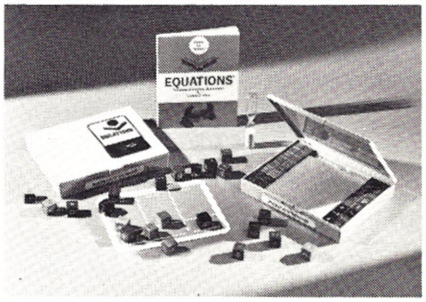
THE PROPAGANDA GAME

2 to 4 players

$11.00*

By Lorne Greene (N.B.C., Paramount Studios), and Robert Allen (Director of Academic Games, Nova University). Students learn the fascinating techniques used by professionals to influence public opinion by learning to recognize "bandwagon" appeals, faulty analogy, out-of-context quotes, rationalization, technical jargon, emotional appeals, and many more. Particularly fascinating for social studies classes, English, problems of democracy, debate.

EQUATIONS



2 to 4 players

$10.00*

The Game of Creative Mathematics, By Layman E. Allen. Our most popular game of mathematics. Five-game kit for use in intermediate grades through high school. Includes arithmetic operations such as addition, subtraction, multiplication, division, exponents and radicals in a variety of number bases. Like chess, the game can be as simple or as complex as the players make it.

ON-WORDS: The Game of Word Structures, by Professors Layman E. Allen and Frederick L. Goodman and Doris Humphrey and Joan Ross, Research Associates (University of Michigan). An instructional game that incorporates any aspect of words that players wish to introduce. Players practice and learn spelling, counting, grammar, phonetics, word roots, inflectional endings, prefixes and suffixes.

2 or more players / For children & adults $10.00*

WFF 'N PROOF GAMES

1490-DF South Blvd., Ann Arbor, MI 48104

____	a. WFF 'N PROOF (logic)	$13.00*
____	b. QUERIES 'N THEORIES (sci. & lang.)	13.00*
____	c. EQUATIONS (mathematics)	10.00*
____	d. ON-SETS (set theory)	10.00*
____	e. PROPAGANDA (social studies)	11.00*
____	f. CONFIGURATIONS (geometry)	6.75*
____	g. ON-WORDS (word structures)	10.00*
____	5-game special (a-e)	49.95*
____	6-game special (a-f)	55.95*
____	7-game special (a-g)	64.95*

*includes postage & handling

My check is enclosed for $______________

Name ________________________________

Address ______________________________

City______________ State______________ Zip________ programs which will diagnose anything. That sort of work requires a tremendous amount of computing capability which we do not have nor are we desirous of obtaining. There are many other institutions in the United States, the University of Missouri at Columbia being one of these, who have done a far better job at that than we will ever be able to do. Our entire thrust is toward the individual practitioner of medicine,dentistry, or veterinary medicine who now has within his reach because of present day technology, the ability to collect, record, and retrieve data in a far more efficient manner than he has ever been able to do before.

Richard E. Easton, M.D.F.A.C.P.M.
President
Health Information Services, Inc.
5341 Parliament Drive
Professional Bldg., Suite 104
Virginia Beach, Virginia 23462

SNOBOL

Dear Editor:

Being a member of the all-too-small group of people who know and love SNOBOL, I was disappointed to see it only barely mentioned in "On Computer Languages" of the September-October issue. Enclosed is a brief description of this mind-blowing language. BASIC may be the best beginner's language in the world, but there *are* problems of a higher nature than SPACEWAR, ya' know.

By the way, PL1 is really spelled PL/I (that's a Roman numeral—IBM's attempt at class). PL/I was supposed to replace FORTRAN, COBOL, assembly language, and everything else, as the special language of the System/360. It is a general purpose language: "good" for lots of things, but not great at anything Math freaks find much more versatility in APL; string manipulation and linguistics people prefer more sophisticated dialects like SNOBOL; artificial intelligence researchers think LISP is best (for LISt Processing, anyway); and operating systems people know that assembly language is far more efficient. "What *is* it good for?" you ask. PL/I is a good replacement for another unimaginative language: COBOL.

Computer Science types also like PL/I, because it allows them to practice structured programming, a technique necessitated by the poor quality of certain languages, notably COBOL, PL/I, ALGOL, and related diseases. I recently watched a group of Intro-to-Programming students hack their way through a PL/I program to do some simple string manipulation. What they did with mountains of DO WHILE's and SUBSTR's and INDEX's could have been done in half a dozen lines of SNOBOL. Likewise, the hairy matrix inversion and statistical problems many FORTRAN students are subjected to could be done in a few lines of APL.

Granted, beginning programmers must learn the concepts of looping, conditional branching, etc. But this can be taught in a more comfortable environment like interactive BASIC. Some of the BASIC interpreters running on minicomputers are really beautiful. And when a student moves on to more complex areas of study, he should be supplied with the appropriate tools. SMALLTALK is an excellent example of an intelligent language for a specialized application. Viva Xerox PARC! I think you'll find that most of the creative programming is done in languages like BASIC and FORTRAN because of their flexibility. COBOL, PL/I, and RPG are for the production-oriented business world, where results are more important than aesthetics.

The high schools with PDP-8's running BASIC/PAL/FOCAL software are still way ahead of the average dp shop with its 360/50 under DOS producing umptween carbons of reports and audit trails no one would even want to read. Computers are mind-expanding tools: programming languages should be designed for this purpose.

I may have made a few readers angry by this time, but isn't that good? Maybe they'll write you about their opinions.

David Touretzky
Rutgers University

An article on SNOBOL may be found on page 32.

A Prejudiced Analysis

Dear Editor:

The "Prejudice Analysis" program by Richard Kahn and Mark Gross described in your September-October issue is a prize example of the misuse of computers denounced by Weizenbaum and others. It falsely claims to analyze, disguises propaganda as science, and intimidates its subjects. To whatever extent Dr. Siegel of Tufts is responsible, he has shown neither the honesty required of a scientist nor the concern for subjects required of a psychiatrist.

Here are the counts of the indictment:

1. **The computer is used to mystify the user, disguise what is being done, and lend scientific authority to opinion.** It claims to "analyze" the extent of a person's racial prejudice. It simply counts the extent of his agreement with the "right answers." The more strongly he holds the right opinions the less prejudiced he is said to be. That cannot honestly be called analysis.

2. **The questionnaire itself is a propaganda document.** Many of the obviously "prejudiced" statements are caricatures of the actual beliefs of most of Siegel's foes — e.g. the opponents of busing.

3. **The program expresses fanatical intolerance.** Each item has a right response, and disagreeing slightly is counted "ANSWERED IN A RACIST MANNER."

4. **The program teaches that objectivity is racist.** Many of the statements concern matters that vary from area to area, e.g. whether a neighborhood is open to Negroes, whether welfare families would soon follow Black families into the neighborhood, whether particular schools would be better with more minority groups, and whether property values would go up or down if minorities entered. The student is taught that circumstances are irrelevant; the non-racist answer is always the same. Many of the questions are only tenuously related to race, but you're racist unless you conform.

5. **The word *prejudice* is misused.** Its use started when it was observed that many anti-Negro and anti-Semitic views were based on hearsay and were usually abandoned on acquaintance. Such a view is a prejudice, but many views called prejudices here are held by people as much acquainted with minorities as Siegel, Kahn and Gross. They still may be mistaken, but they are not prejudices unless thoughtlessly held.

6. **The program instigates and manipulates guilt feelings to browbeat students into conformity.** The student soon recognizes that he is in the hands of fanatics. If he wants the good will of his teacher and his right-thinking classmates, he will know how he must answer all but question 9. Unless he is rather subtle, he will believe he ought to disagree with "*City riots are a threat to our suburban life*" just as he must disagree with the more traditional "*Property values will go down if minorities enter the neighborhood.*" But this is the racist response.

Perhaps not agreeing doesn't take rioters seriously enough. The effect is to trip the unwary and worry the wary. Almost no subject will get a clean bill of health, and each will go away with a feeling of guilt.

7. **Using the program in a classroom violates the civil right of a student not to be swindled and browbeaten by his teachers and university researchers.** Considered as a scientific experiment involving human subjects, it violates every pertinent code of ethics and may violate Massachusetts or Federal law.

I fear that such abuse of questionnaires and computers has become widespread in the social sciences, but it is rarely described as clearly and concisely as Messrs. Gross and Kahn have done. Since the program is dated 1970, one can hope they have recovered some objectivity, honesty and fairness. Still it would be interesting to know how many high school classes were subjected to this computerized indoctrination.

The reader may think it foolish to have put even this much effort into attacking a high school students' hack. Maybe no class was ever subjected to it, and if one was, maybe the students were not intimidated, thought it was silly, and said so. My excuse is that it is rare and refreshing to find all these common intellectual crimes concentrated in two pages and a computer program whose unambiguous behavior leaves no room for the authors to claim they were misunderstood.

John McCarthy
Computer Science Department
Stanford, CA 94305

We Design

Educational Computer Systems

School Computers are special.
More than hardware is necessary.
Instructional programs must exist
in a wide range of disciplines.
Administrative software must be created
for unique requirements in education.
Students and administrators should be able
to work simultaneously and independently.
Operating systems must provide excellent security.

Most importantly, the company providing the
system must be an educational specialist.
Be able to determine your unique needs,
funnel information from other users,
train people, on-site if necessary,
create new educational software,
and provide continuous support
even after the installation.

A FIVE (5) TERMINAL TIMESHARING SYSTEM WITH DISKETTE STORAGE IS UNDER $20,000.

COMPLETE SYSTEMS START AT $5,915.

Please call me about the following system:

□ Low Cost
□ Medium Scale Timesharing
□ Administrative
□ Instructional and Administrative

Name____________________ Title____________

School________________________________

Address_______________________________

City____________________ State____________

Phone___________________ Zip____________

educomp
corporation
196 Trumbull Street
Hartford, Connecticut 06103
(203) 728-6777

al... editorial... editori

We are asking you, dear readers, to indicate your reaction to various articles and features in *Creative Computing.*

Our purpose is twofold: to learn which articles from our issues you read together with your constructive criticisms, and to find out more about who you are. This information is vitally important to us and your time in filling out our form will be most appreciated. Why?? Read on.

As a result of informal surveys in the past we have some ideas about which articles will be most read when we put together an issue of *Creative Computing*. For example, we know that the "Complete Computer Catalogue" section is more widely read than the longer reviews even when they carry substantially similar information. We know that articles about programming techniques in BASIC will be read more than, say, an article on medical applications.

Yet, we cannot always choose articles that will have the highest initial readership acceptance. We feel that you demand from us leadership in exploring new, creative uses of computers. We believe that you share our concern that computers be made to benefit people and help shape a more humane future, and that we must pay attention now to the directions in which they are developed. We feel it is incumbent upon us to expose our readers to unfamiliar subjects that may help stretch their thinking, as that exposure has helped broaden our vision. To cover these areas in sufficient depth, it may be rough going for the uninitiated. Microprocessors, for example, are of increasing importance to everyone whether they realize it or not. We shall keep pounding away.

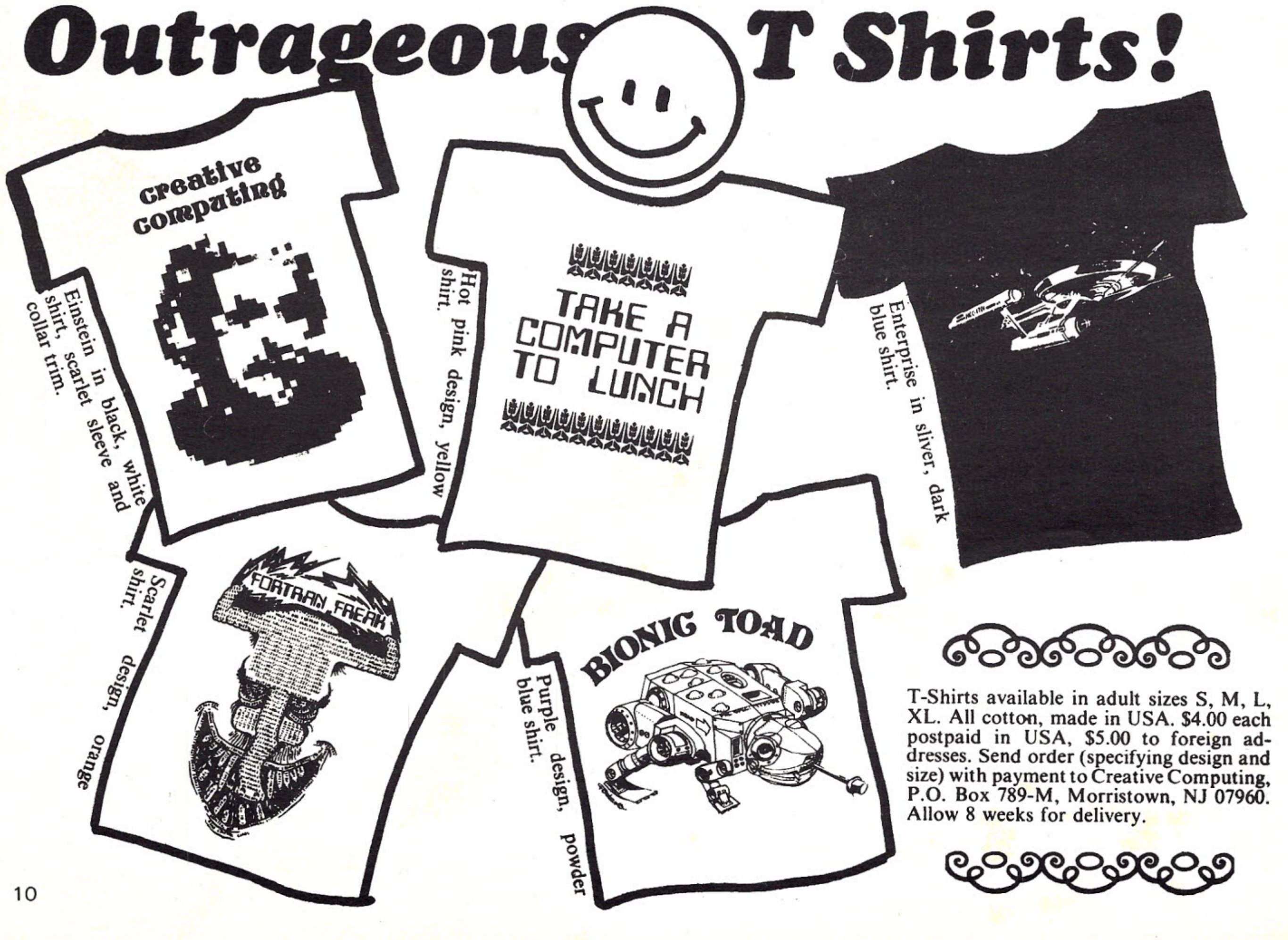
Nevertheless, we certainly don't have all the answers, nor are we so bullheaded that we won't accept comments or criticisms from our readers. For if, regardless of our purpose to help inform you as you become a leader toward a better future, you're just not reading our stuff, period, our grandiose ideas are for nought. So please, now, vent your passions, tell us what you'll read.

The reason for asking about who you are is a simple old-fashioned matter of economics. We want to have more advertising in *Creative Computing* to help defray expenses so that we can afford to increase the size and printed quality of the magazine, afford more top quality work, pay the increased postal rates, paper rates, etc., and *not increase the cost of subscriptions* in this day of inflation. We could do it with advertising, but many potential advertisers want to know more about you, our subscribers, before they'll give us their ads. Also, we think we're asking for ads from people whose products will be interesting to you, which would be an attraction to, rather than a detraction from, our magazine. So how about it, will you help us out?

David Ahl

Please Return Form

Outrageous T Shirts!



Einstein in black, white shirt, scarlet sleeve and collar trim.

Hot pink design, yellow shirt.

Enterprise in sliver, dark blue shirt.

Scarlet design, orange shirt.

Purple design, powder blue shirt.

T-Shirts available in adult sizes S, M, L, XL. All cotton, made in USA. $4.00 each postpaid in USA, $5.00 to foreign addresses. Send order (specifying design and size) with payment to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. Allow 8 weeks for delivery.

CREATIVE COMPUTING SURVEY

We hope that you are motivated through lofty ideals and noble purposes to respond to our questionnaire. But in the event you demand a crass, greedy, materialistic incentive, here it is. We are having a drawing of all submitted survey forms. The first prize winner receives a 3-year subscription or renewal of *Creative Computing*. The second and third place winners each receive a one-year subscription or renewal.

WIN A 3-YEAR SUBSCRIPTION TO CREATIVE COMPUTING, OR ONE OF TWO 1-YEAR SUBSCRIPTIONS!

Complete this form and return it for the drawing. Deadline is January 2, 1977.

A.

1. In an issue of *Creative Computing* I read:	Always	Mostly	Some times	Never
Feature Articles	□	□	□	□
Book Reviews	□	□	□	□
Fiction	□	□	□	□
Puzzles and Problems	□	□	□	□
Simulations and Learning Programs	□	□	□	□
Game Programs	□	□	□	□
Letters to the Editor	□	□	□	□
Advertisements	□	□	□	□
Editorials	□	□	□	□
Compleat Computer Catalogue	□	□	□	□
Cartoons, Humor	□	□	□	□
Notices	□	□	□	□
2. I use the games or programs listed:	□	□	□	□

B.

1. I would be interested in more articles about:	Very Much	Some	Not Much	Waste of Space
Building a Computer	□	□	□	□
Microcomputers	□	□	□	□
Computer use in Education	□	□	□	□
Computer Applications in:				
Music	□	□	□	□
Art/Graphics	□	□	□	□
Medicine	□	□	□	□
Space Exploration	□	□	□	□
Business and Industry	□	□	□	□
Home Control	□	□	□	□
Computer Graphics	□	□	□	□
Artificial Intelligence	□	□	□	□
Learning Activities	□	□	□	□
Fiction	□	□	□	□
Social Implications	□	□	□	□
Computer Club Activities	□	□	□	□
Other__________				
2. I would like more technical articles:	□	□	□	□
3. I would like more reprints from conference talks:	□	□	□	□

C. My function is:

□ Faculty: College or University
□ Faculty: Grades K-12
□ Student: College or University
□ Student: Grades K-12
□ Industry: title__________
□ Government: title__________
□ Other (please specify)__________

D. I utilize a computer at (check all that apply):

□ work
□ school
□ home
□ Other__________

E. At home I have (check all that apply):

□ microcomputer
□ minicomputer
□ CRT terminal
□ hard copy terminal
□ floppy disk
□ digital cassette tape
□ video display
□ Other__________

F. I utilize my home computer for:

□ recreation
□ CAI
□ research
□ record keeping
□ home control
□ work related

G. (If in education) I use a computer:

□ in a computer center
□ in my discipline which is__________

H. My age:

□ under 20
□ 21-35
□ 36-50
□ 50 or over

I. Books I read are: (check all that apply)

□ science fiction
□ modern literature
□ sports
□ mystery
□ history
□ adventure

J. Other magazines I read are: (check all that apply)

□ Byte
□ Communications of the ACM
□ Computer
□ Computer Decisions
□ Computerworld
□ Curriculum Product Review
□ Datamation
□ Dr. Dobb's Journal
□ Interface
□ The Mathematics Teacher
□ Mini Micro Systems
□ Minicomputer News
□ People's Computer Company
□ Science News
□ Scientific American
□ The Journal

K How many people read this copy of *Creative?*

□ one
□ two
□ three
□ more?__________

L. I save my copy:

□ a week or two
□ a month or two
□ keep it (file it, save it, treasure it)

M. I recommend, specify, select or purchase: (check all that apply):

□ Medium/large computer
□ Minicomputer
□ Microcomputer
□ MPU chips
□ CRT Terminal
□ Teleprinter
□ Graphics Terminal
□ Coupler/Data set
□ Peripherals
□ Programmable calculators
□ Hand calculators
□ Computer leasing service
□ Software
□ Courseware
□ Learning Aids
□ Books/Publications
□ Consultants
□ Other__________

N. Please add comments: __________

Name __________
Address __________
City __________
State & Zip __________

Return to: Creative Computing, P. O. Box 789-M, Morristown, N.J. 07960

COMPLEAT COMPUTER CATALOGUE

We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," *Creative Computing*, P.O. Box 789-M, Morristown, NJ 07960.

BOOKS AND BOOKLETS

CALCULATORS IN THE ELEMENTARY SCHOOL

A 59 page special report published by the Curriculum Group of the Oregon Council for Computer Education. The first half of the booklet contains many calculator games and exercises. (To find out what Amelia Earhart's father said the first time he saw her flying an airplane by herself, Find .023 x 3, add 10141 to the result, multiply by 5, and look at the answer upside down.) The rest of the report describes a six-week experiment using calculators with fifth and sixth grade students including an outline of each day's activities, the results, and conclusions. Two annotated bibliographies. $2.00.

Oregon Council for Computer Education, 4015 SW Canyon Rd, Portland, OR 97221

COMPUTER USES IN EDUCATION

Proceedings of the ACM SIGGSE-SICCUE Symposium in Anaheim, California, in February 1976 are available in a 400 page publication. If you're interested in these important papers describing innovative uses of computers in education get the book. It's doubtful you'll get a chance to read them anywhere else since reprint costs run $500 an article. (Let the ACM know how you feel about that too!) Price for ACM or SIGGSE members is $15 prepaid; others $20 prepaid.

ACM Order Department, P.O. Box 12105, Church Street Station, New York 10249

NO FREE WILL IN TOMATOES

This is the title of a little (2¾ x 4½"), handmade (sewn and tied), limited edition (300) chapbook of 14 "minimal poems" by Peter Payack. These witty poems, some of which have previously appeared in *Creative Computing* show Peter's fascination with history and science.

Peter Payack and Jane Barnes publish chapbooks under the name of Quark Press (A quark being the smallest identifiable physical particle known today). Peter is also founder of Phone-A-Poem in the Boston area (call 617-492-1144 for a delightful short poetry reading by a contemporary poet).

Three chapbooks are in Quark's First Series: *Curios* by F.A. Nettelbeck, *Mythologies* by Jane Barnes and *No Free Will in Tomatoes*. $1.00 each, $2.50 for all three.

Quark Press, Box 193, Cambridge, MA 02141

COMPUTER GAMES IN MACHINE CODE

Scelbi's First Book of Computer Games contains three games in machine code for 8008 and 8080 based systems: "Space Capture," Hexpawn, and Hangman. Hexpawn and Hangman are modeled on the familiar games, and the idea in Space Capture is to shoot down a computer-controlled spaceship roaming the galaxy. Scelbi also has another game book called Galaxy. In Galaxy the object is to search through a galaxy of 64 quadrants (each composed of 64 sectors) to find alien ships and destroy them with torpedoes or "phasors." Sound familiar? Anyway, both books include complete programs, illustrations, and flowcharts. Scelbi's First Book and Galaxy are both $14.95 apiece, ppd.

Scelbi Computer Consulting, Inc., 1322 Rear Boston Post Road, Milford, CT 06460

HP EDUCATIONAL SYSTEMS INFORMATION

Hewlett Packard, one of the most active computer vendors in the educational arena has announced a number of new applications and software packages further enhancing their hardware for educational customers.

The 12-page booklet "Computer Solutions for Elementary/Secondary Schools" describes hardware, software (instructional, CAI, administrative, etc.) for the HP 2000 and 3000 series. Free.

"Computer Solutions for Colleges and Universities" is a 12-page booklet emphasizing software in instruction, administration, and networks as well as user services. Free.

"HP Math" describes three CAI math courses for students in grades 1 to 6, the higher grades, and for adults who need remedial work in arithmetic. Free.

The 128-page "HP Clearinghouse" catalog describes almost every known application of HP minicomputers in education. It contains 80 entries on instructional and administrative applications, 50 entries on utility packages, and 90 descriptions of books, catalogs, and periodicals. $2.00.

"Learning Timesharing BASIC" is a 60-page booklet designed to teach BASIC to beginners in a light and easy-to-understand way. $3.00 [A fun text, although we feel you can do better; see *Creative's* review of 34 books on BASIC in previous issues.]

Inquiries Manager, Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, CA 94304

CALCULATORS IN THE CLASSROOM

A 24 page report on a symposium sponsored by Rockwell International in December 1974. Most of the controversy centers around whether or not the use of calculators in the classroom is justified or whether it is just a passing fad (as "language labs" once were). Price unknown.

Arnold Isford, Manager, Educational Marketing, Microelectronic Product Division, Rockwell International, 3310 Miraloma Avenue, PO Box 3669, Anaheim, CA 92803

"Knowledge is power."

Bacon

ORGANIZATIONS

SCS

The Society for Computer Simulation is the principle technical society devoted to the advancement of simulation through the use of computers and similar devices. SCS seeks to promote the development of simulation technology through the exchange of information among people who use simulation to advantage in their endeavors. Full membership is open to anyone who has been in any phase of simulation for at least four years. Membership is $25/yr, students $15.

Alex McKenna, The Society for Computer Simulation, P.O. Box 2228, La Jolla, CA 92038

ACM

The purpose of the Association for Computing Machinery is to advance the sciences and arts of information processing including the study, design, development, construction, and application of modern machinery, computing techniques, and languages. Persons qualified to be ACM members must subscribe to its purposes, have attained professional stature as demonstrated by intellectual competence and ethical conduct in information processing, and must be endorsed by two members of the ACM. Student membership also available at reduced rates. Many diverse publications and Special Interest Groups.

Joseph Cunningham, Association for Computing Machinery, 1133 Avenue of the Americas, New York, N.Y. 10036

ACM SIGCSE

The purpose of the ACM Special Interest Group on Computer Science Education is to provide a forum for problems common among college educators attempting to develop, implement, or evaluate computer science programs, courses, and problem sets. Quarterly newsletter. Non-ACM membership in SIGCSE $9.00/yr.

SIGCSE, ACM, 1133 Ave of the Americas, New York, NY 10036

ACM SIGCUE

The Special Interest Group of Computer Users in Education of the ACM has as a major purpose the interchange of information among educational (instructional) users. The membership of 1200 or so is skewed toward higher education. The quarterly bulletin is substantial (40 typed pages) and carries in-depth articles, interviews, and conference reports. Each issue also runs an annotated bibliography of new books and magazine articles. The group sponsors meetings at the annual ACM, NCC, and CCUC conferences, and a joint conference with SIGCSE (first time in 1976). Annual membership $6 for non-ACM members.

SIGCUE, ACM, 1133 Avenue of the Americas, New York, NY 10036

ACM SIGCAS

The purpose of the ACM Special Interest Group on Computers and Society is to provide a forum for the examination of the impact of computers on society, in terms of major economic, political, and sociological trends; information systems in many areas; privacy; and related issues. Very professional quarterly newsletter. Non-ACM membership in SIGCAS $9.00/yr.

SIGCAS, ACM, 1133 Ave. of the Americas, New York, NY 10036

IEEE COMPUTER SOCIETY

The IEEE Computer Society is actually part of a much larger organization, the Institute of Electrical and Electronics Engineers. It was formed to advance the theory and practice of computer and information processing technology and exchange technical information among its members. These people are concerned mostly with hardware. To be eligible for membership in the IEEE Computer Society you must be an IEEE or approved society member, have graduated from a four-year course of study, have been involved in the computer field professionally for at least five years, or be a registered student in the Society's field of interest Membership including IEEE dues $46/yr.

Harry Hayman, P.O. Box 639A, Silver Spring, Maryland 20901

ASCUE

This organization of Small College Users in Education aspires to encourage the appropriate uses of computing equipment and techniques for its member institutions and to assist its members in solving individual problems. Formerly an IBM 1130 users group, member institutions today have DEC, HP, GA, and other hardware as well as the inevitable IBM systems. Regular Membership is $35 for educational institutions.

Dan Kinnard, ASCUE Director of Public Relations, New Mexico Military Institute, Roswell, New Mexico 88201

COMNET

ComNet—Northeast Information Network is a cooperative resource and skills exchange for Washington, Oregon, Idaho, and British Columbia. It serves as a community memory bank providing people with information on positive options and alternatives they need for now and the future. ComNet makes information available on topics such as alternative sources and uses of energy, communications, food & nutrition, tools and low impact technology, and transportation. Individual membership for a year is $6.00 and includes 4 issues of the magazine Northwest Synergy Access. Include a SASE.

ComNet Northwest Information Network, Box 5599, Seattle, Washington, 98105

ADCIS

The Association for Computer-Based Instructional Systems focuses on CAI in general, with particular emphasis on educational psychology and applications in the medical sciences. It publishes a newsletter which carries information about the activities of individual members and abstracts of the association's two meetings each year.

Peter Dean, Box 1403, Los Gatos, CA 95030

AEDS

The Association for Educational Data Systems was founded in 1962 to provide a forum for the exchange of ideas and information about the relationship between technology and modern education. Primary emphasis tends to be on public school data processing applications with instructional applications taking a back seat. Individual membership $20, student $10. Membership is "open to all interested in learning more and keeping informed about current developments in educational data systems." AEDS also holds an annual convention and sponsors a programming contest for students in grades 7-12.

Shirley Easterwood, Association for Educational Data Systems, 1201 16th St. N.W., Washington, D.C. 20036

NAUCAL

The National Association of Users of Computer Applications to Learning is an organization of users whose purpose is to improve the learning-teaching process by influencing the development, evaluation, and dissemination of computer applications to learning. NAUCAL provides a forum for the exchange of ideas in the areas of drill-and-practice, tutorial, simulations, problem solving, computer-based testing, and other facets of instructional computing. It was founded in 1970 and is now a functional chapter of AEDS. Membership is on an individual basis at $5.00 per calendar year.

Mr. George H. Litman, Secretary/Treasurer, Board of Education, City of Chicago, 228 North Lasalle Street, Room 430, Chicago, Illinois 60601

ASIS

The American Society for Information Science is a nonprofit professional association organized for literary, scientific, and educational purposes and dedicated to the creation, application, and dissemination of knowledge concerning information and its transfer. The Society acts as a bridge between research and development and the requirements of diverse types of information systems. Annual conference with proceedings, bimonthly Journal of the ASIS, 10 times/yr Bulletin. Regular membership $35, student $10.

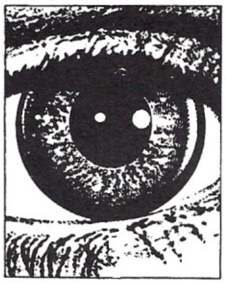
Robert McAfee, Jr., American Society for Information Science, 1155 16th Street, N.W., Suite 210, Washington, D.C. 20036

MEDIA

COMPUTER ART SLIDES

The Computer Arts Society has collected a set of 35mm slides of artwork done by computer. Sets of approximately 100 slides are available at cost (approx. 30¢ per slide, i.e. $30.00/set).

Ruth Leavitt, 5315 Dupont Ave South, Minneapolis, Minnesota 55419.



EYES FOR COMPUTERS

Robotics and machine intelligence are explored in this film from the General Motors Film Library. Shown are attempts to develop decision making powers into the computer—decisions based on what the computer "sees" via TV cameras. GM's object is to develop machines to do some assembly line work, inspection procedures, and "intelligent" parts handling. The problems associated with a simple task such as recognizing a three dimensional object and then doing something with it are immense, as suggested by this film. This is an interesting computer applications film. It's short, crisp, and good! About 10 minutes. Free.

General Motors Film Library, Detroit, Mich. 48202

VIDEO TAPE ON MAGIC SQUARES

A 30 minute color video tape describing the magic squares of De La Loubere and Franklin has been produced by Donald T. Piele at the University of Wisconsin-Parkside. The tape contains: a brief historical description of the De La Loubere and Franklin Magic Squares; an explanation of the algorithm used to construct each square; and a segment showing the magic squares being generated by computer. The tape is appropriate for junior high school and above, and is available in many forms: ½" reel - $25, ¾" reel or cassette - $30, 1" reel - $30. If you send your own reel or cassette deduct $15.

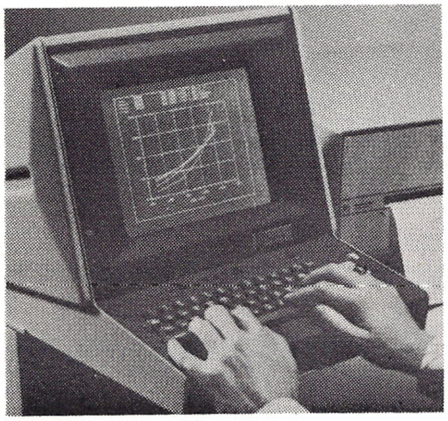
David Campbell, Director, Media Production Center, University of Wisconsin - Parkside, Kenosha, Wisconsin 53140

FATHER OF THE COMPUTER

"Charles Babbage, Father of the Computer" is a 28-minute color TV documentary which introduces Babbage's amazing calculating engines and many basic concepts of computer technology. It is available in the four popular TV tape formats (videocassette, IVC 1", EIJ ½", and broadcast standard 2"). Loan of tape, free; dubbing at reasonable rates.

AFIPS History of Computing Project, 20 Wilson Road, West Point, NY 10996.

HARDWARE



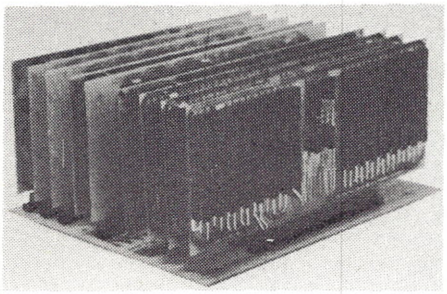
TEKTRONIX GRAPHICS PRODUCTS

Following close on the heels of interactive computing in schools is interactive graphics. And few vendors have a more comprehensive line of graphics products than Tektronix ranging from the inexpensive 4006 graphic computer terminal ($2995) to the 4014 big 19" screen terminal ($10595) to the 4051 stand-alone BASIC graphic computer ($6995).

Worthwhile free literature from Tektronix includes Tekgraphics (April 73, No. 5, "Educational Applications"), 4051 Flyer, Computer Products Catalog, and Price List.

Tektronix, Inc., Information Display Group, P.O. Box 500, Beaverton, OR 97077

[See the complete review of the 4051 on pp. 20-21.]



THE DIGITAL GROUP

The Digital Group has a new idea. Namely a microcomputer system with interchangable CPUs at the CPU card level. In other words, a system doesn't become immediately obsolete with each new microprocessor announcement. Your major investment in memory and I/O is protected.

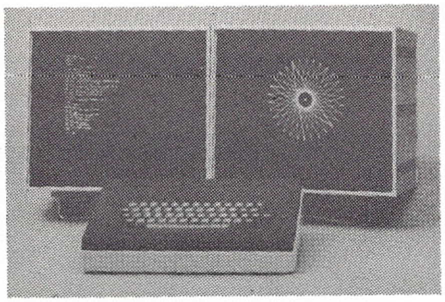
A typical Digital Group System consists of a CPU card with 2K RAM (choice of Zilog/Mostek Z-80, Intel 8080A, Motorola 6800 or MOS Technology 6502), I/O card with four input and four output ports, TV Readout and audio cassette interface, 8k static RAM and Mini-Mother board. You have to add a power supply, ASCII keyboard, cassette recorder, TV set and cabinet (if you wish) and you're ready to go.

Currently available software includes Tiny BASIC, a number of Tiny BASIC games, some ham and educational packages.

Kits with 2k memory (no power supply) start at $375 (6502), $425 (8080,6800), or $475 (Z-80). Add $135 for 12A power supply, $225 for 8k memory and keyboard (around $50) and you're in business. More information is free.

[Incidentally, The Digital Group has gained a good reputation for not announcing products before they can be delivered — DHA]

The Digital Group, Inc., P.O. Box 6528, Denver, CO 80206



TT250 GRAPHICS TERMINAL

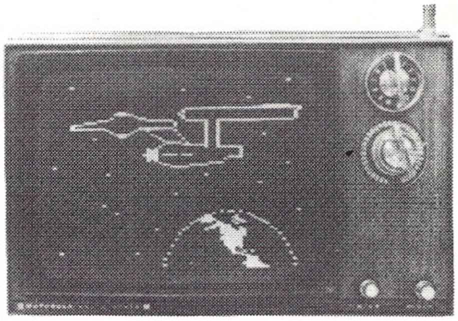
Developed at the MIT Artificial Intelligence Lab, this terminal is different from any other you've probably seen since it uses two screens! One is used for text, and the other for *dynamic graphics*. The characters are on programmable 16x8 fonts with no spaces between the characters—so you can combine the characters to form larger symbols or even diagrams. In normal use the separation between characters is treated as part of the character itself. Also, the graphics have the capability of animation; that is, they can be changed smoothly and rapidly. The TT2500 has its own built-in processor and starts at $5,950.

General Turtle Corporation Limited, 120 Boul. Industriel, Boucherville, Quebec J4B 2X2, Canada

CRT TERMINALS

DS-3000 KSR terminal from Hal Communications can receive and send Baudot (5 level) and/or ASCII (8 level) code data. Built-in microprocessor allows full cursor positioning and editing of the full screen of 16 lines x 72 characters/line. Hal seems more interested than most manufacturers in schools and hobbyists.

HAL Communications Corp., 807 East Green St., Urbana, IL 61801



LOW-COST GRAPHICS TERMINAL KIT

The GT-61 Graphics Terminal Kit, produced by Southwest Technical Iroducts Corporation, displays an array of cells 64 wide by 96 high on a standard video monitor or modified TV set. Each cell can be selectively turned on or off by the computer. There is also a provision for mixing graphics and alphanumerics on the same screen when using the GT-61 with Southwest's CT-1024 TVT. And since the GT-61 may be driven by any computer having a TTL compatible 8 bit parallel output, it isn't necessary to own a particular system to use it! The GT-61 is $98.50, less power supply, chassis, and TV monitor.

Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, TX 78216

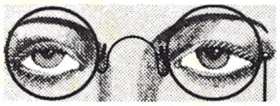
IBM 5100

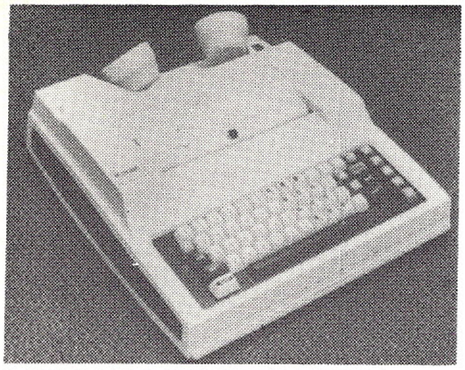
The 5100 is IBM's first entry in the true minicomputer/calculator field. This desktop unit, slightly larger than an electric typewriter, contains CPU, keyboard, 4" CRT display, cartridge tape unit, memory (up to 64k) and can run BASIC, APL, or both. The 5100 can drive a TV monitor directly. Options include a telecommunications interface, medium-speed printer, and an auxiliary tape unit. (BASIC takes 4k overheard, APL 6k). Prices are as follows:

Memory	BASIC	APL	Both
16k	$ 8,975	$ 9,975	$10,975
32k	11,975	12,975	13,975
48k	14,975	15,975	16,975
64k	17,975	18,975	19,975

Printer is $300, aux tape $2300, comm. adapter $900, programming packages (business, stat, math) $500 each, BASIC CAI pkg $225, APL CAI pkg $295.

IBM, General Systems Div., P.O. Box 2150, Atlanta, GA 30301

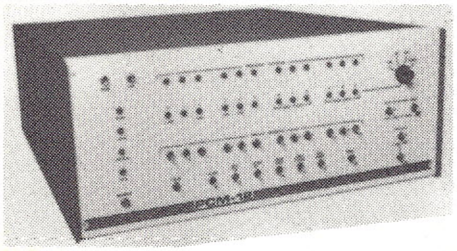




LIGHTWEIGHT TERMINAL

Weighing only 13½ lbs including coupler and carrying case, the Tymshare Model 125 terminal requires only a standard telephone and electrical outlet for fully portable KSR operation. Features include silent printing, color-coded keyboard, integrated 13-key numeric pad, and alternate 30 or 10 cps operation. Sale price: $2210; rental as low as $100/month. The Model 125 is available through Tymshare offices in major cities in the United States.

Tymshare, Inc., 20705 Valley Green Drive, Cupertino, CA 95014



PCM-12 COMPUTER KIT

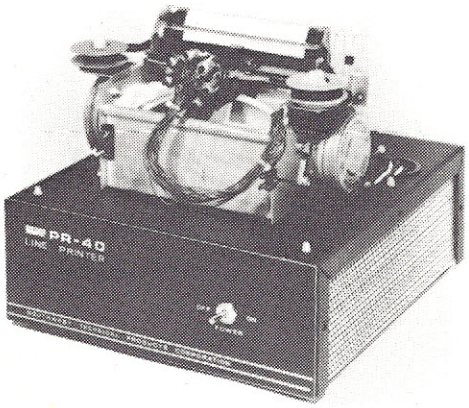
If you're thinking about buying a computer kit, then this one is worth some consideration. The PCM-12 is a simulation of a PDP-8 based on the Intersil IM6100 microprocessor. It runs the same software and has essentially the same front panel functions as the PDP-8. As a result there is already a ton of cheap software for the PCM-12. When ordered with the kit, 4K BASIC on papertape is only $1.50! The basic system with 1K of memory is $799. An 8K system with terminal interface (but no terminal) would run $1417.

PCM, Box 215, San Ramon, CA 94583

TALKING CALCULATOR

With a 24-word vocabulary, the SPEECH-PLUS calculator enunciates—in easy-to-understand electronic speech—each keystroke as it is made and then announces the answer. In production applications, it is useful to verify entries without watching the visual display and for education it is valuable to capture and hold the attention of students. Four functions, square root, percent, independent memory, constant, floating decimal, eight-digit capacity. Earphone jack for private listening; operates 3 hours between charges. Instructions available in print, braille, or cassette. Complete with case, earphone, battery charger and instructions $395.00.

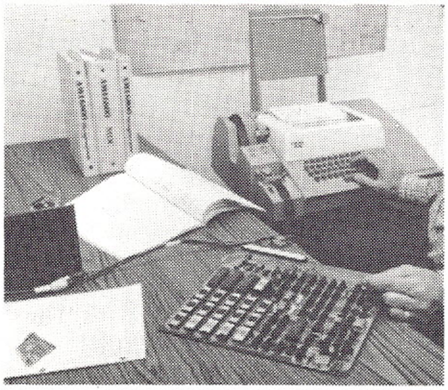
Telesensory Systems, Inc., 1889 Page Mill Rd., Palo Alto, CA 94394



LOW COST PRINTER KIT

Southwest Technical's PR-40 Alphanumeric Printer Kit uses a 5x7 dot matrix impact print mechanism. It prints the 64 upper case ASCII characters, 40 characters per line, 75 lpm on standard 3⅞" rolls of adding machine paper. Complete kit $250.00.

Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio, TX 78216



PROTOTYPING BOARD FOR AMI 6800

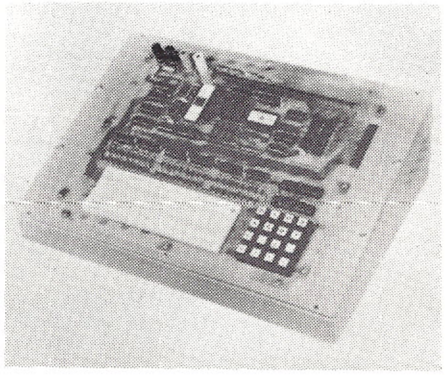
This board is useful for hardware and softward evaulation of 6800-based microcomputer applications. It contains a built-in programmer for the S6834 EPROM and Tiny BASIC is also available on the EPROM at no extra charge. Communication to the outside world is done with a TTY. A minimum kit is $295 and a fully tested unit with Tiny BASIC on EPROM is $950.

American Microsystems, Inc., 3800 Homestead Rd., Santa Clara, CA 95051

JOLT

JOLT is a microcomputer system based on the 6502 microprocessor with a debugger/monitor on a special read-only memory package called DEMON. This firmware permits use of any terminal from 10-30 cps and lets you display CPU register, memory locations, also load from and write to papertape. JOLT is not a mainframe, it's one of those things where the boards stack one on top of another. The CPU board is $249 and other options are available.

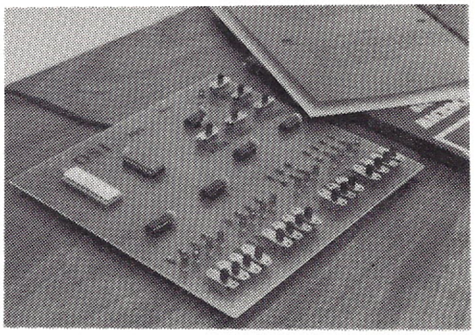
Pehaco Corporation, Jolt Sales Agents, Microcomputer Associates Inc., Dept. A, 111 Main Street, Los Altos, CA 94022



MINI-MICRO DESIGNER

The MMD-1 is an inexpensive educational microcomputer with easy breadboarding capabilities. It has a keyboard entry system and is intended for use at the machine language level and not for sophisticated applications. The MMD-1 is based on the 8080A chip and has on-board PROMs to control keyboard entry. A complete MMD-1 kit is $350. Bugbook III provides extensive documentation including 60 experiments for the MMD-1 in its 592 pages. $14.95.

E & L Instruments Incorporated, 61 First Street, Derby, Connecticut 06418



COMPUTER TRAINER

The Model 300 Computer Trainer is a completely assembled and tested, ready-to-use computer which comes complete with a 20 experiment lab manual written for use with a college physics, electronics, or computer course. Since the manual assumes no knowledge of computers or digital electronics, it is also ideal for self-teaching. The first lab is simply loading and reading memory, and the last experiment is interfacing the computer to a TTY.

The unit comes with 128 words of memory and is based on the MOS Technology 6502 chip. The Trainer with lab manual is $99, power supply is $10, and hardware and programming manuals are $10.

Ohio Scientific Instruments, Box 374, Hudson, Ohio 44236

CALCULATORS

ASKING THE CALCULATED QUESTION

NOVUS presents a calculator for the older siblings of the young kids who use that wise old owl QuizKid calculator described in the Sept/Oct catalogue. It's the more serious, more versatile QuizKid II. This lets the user choose a type of problem (addition, multiplication, subtraction or division) and a desired speed, then presents a timed series of ten problems. The child gets two tries at the correct answer after which it is displayed. The score is revealed at the end of the series. With a game adapter and over 1200 problems it's a hoot of a value at $24.95.

To order and for information on QuizKid II and other exciting calculator and mathematic learning aids contact: Mr. Jay Hemming, Educational Marketing Manager, National Semiconductor, 1177 Kern Avenue, Sunnyvale, CA 94086.

BIORHYTHM CALCULATOR

If you're one of those biorhythm freaks then you'll be interested in a new Casio calculator called a Biolator, which in addition to performing regular mathematical functions, also computes your biorhythm. It has a 99 year calendar which permits you to calculate the number of days since your birth (if it was in this century—tough luck, senior citizens). The biorhythm is supposedly a representation of a person's physical, emotional, and intellectual states. Suggested retail price is $29.95. Look in a local department store because we don't have Casio's address.

MISC.

TECHNOLOGY EXCHANGE SERVICE

Technotec is a system available from over 5000 computer terminals in more than 150 metropolitan areas worldwide. Its purpose is to get together those who have useful inventions or know-how with those seeking to acquire new technology, in the form of a "Techno-Stock." Techno-Stocks in a particular area of interest can be located easily by entering keywords. Technotec isn't an information system, but is a communication system built around a central data base. A general information manual is $3.50 (Order No. 76073300).

Control Data Technotec, Inc., Box O, Minneapolis, MN 55440

A Creative Computing Equipment Profile . . .

TEKTRONIX 4051 GRAPHICS SYSTEM

by Stephen B. Gray
Gray Engineering Consultants
Darien, CT

The Tektronix 4051 is one of several hard-wired BASIC computers (which can be programmed only in BASIC), but it is the only one that also offers graphics, as well as the ability to function as a terminal. I had the good fortune to be able to borrow a 4051 for a month, and I'd like to tell you about this fascinating desk-top (or pedestal-mounted) graphics computer. First, a look at the hardware, then later the software.

The Hardware

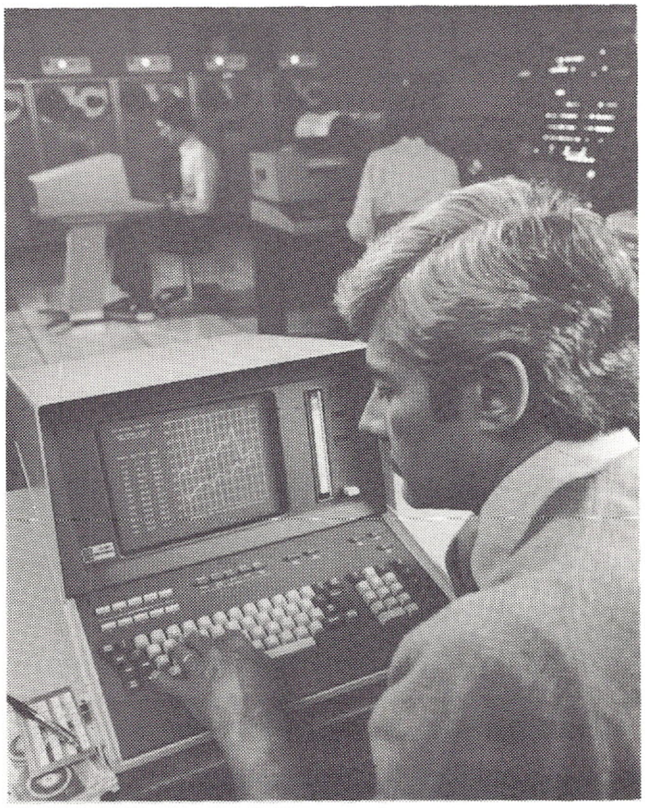
The main keyboard offers the full 128-character ASCII set with both upper and lower case; 96 printing characters and 32 control characters. Several additional built-in character fonts permit the use of accented vowels and special signs such as the British pound sign. To the right of the main keyboard is a numeric keypad which, when used with nearby keys for the four arithmetic functions plus parentheses and exponentiation, permits fast calculator-type computations, without line numbers or any programming at all. The keypad is also a help when entering numeric data into a BASIC program.

A group of ten user-definable keys, used in conjunction with the shift key, allows up to 20 pre-defined program subroutines to be individually selected for injection into the main program. The first 99 line numbers are reserved for these User Definable keys; each key transfers program control to the first of four line numbers, enough for a *very* short subroutine or, more likely, enough for a GOSUB to a larger subroutine. If the last of the four lines is not a RETURN (or END or STOP), the system keeps executing statements in sequence, in the user-definable routines, until it finds a RETURN, STOP or END, or until it continues into the main program beginning at line 100. These keys are very handy for menu selection, and a plastic overlay card you can write on keeps track of what you've got them programmed for.

At top center of the keyboard, five keys provide ten functions for program editing, permitting characters to be changed, deleted or added. A nearby Auto Number Key will provide a line number automatically for each BASIC statement before it is entered on the keyboard, starting with 100 and incrementing by 10; the initial line number and increment can be changed to whatever is desired. A Step Program key permits executing the current BASIC program one step at a time, starting at the first line or, by using a GOTO, at any line desired.

At top right are three keys for peripheral control. The Auto Load Key causes the internal magnetic-tape unit to load the first program found on the current tape cartridge. The Rewind key will rewind the tape cartridge back to the beginning, and Make Copy causes the optional Hard Copy Unit, if one is attached, to make a paper copy of the information on the display.

The vertical slot at the right of the display holds a 300-kilobyte magnetic-tape cartridge; more about that when we get into software.



The display uses an 11-inch-diagonal direct-view storage crt, with 72 characters per line, 35 lines per page, 1024 x 780 addressable graphic points. Clever feature: the data on the screen dims to a lower (but still readable) level after a few minutes, to conserve energy; pressing the shift key restores the original brightness.

The MPU used in the 4051 is a 6800; a 32k ROM holds the operating system. The 4051 is supplied with an 8-kilobyte RAM for work space; memory is expandable in 8k options up to 32k maximum.

Hardware options include the 4631 Hard Copy Unit for dry-process copies; 4924 Magnetic Tape Unit for 300k *external* bytes; 4662 Digital Plotter for X-Y plotting and digitizing on a 10-by-15-inch work area; 4952 Joystick for positioning the graphics cursor in interactive graphics.

Communications Option

A communications option, which adds an RS-232C interface, lets the 4051 act, in terminal mode, like a Tektronix 4012 Computer Display terminal, with keyboard inputs going direct to whatever computer the 4051 is connected to, and with returning data and graphics going up on the screen. The same option lets the 4051, in communications mode, use the internal tape unit to send or receive data at speeds up to 2400 baud. An output-only RS-232C interface, for line printers, is also available. Either option is in the form of a plug-in ROM package, about the size of two cigarette packs, which fits into a receptacle in the back of the 4051. There are three others, which I haven't seen: matrix, editor, and binary-loader ROM packs; more are in the works.

Software and Manuals

Four manuals and two pre-recorded tapes come with the 4051. The two manuals on BASIC will be covered in a future review. The tapes are duplicates; one is to be put "in a safe place; it exists to minimize down time in the event the operational tape is accidently erased." Each tape contains these programs: system verification, tutorial, Y-only data plot, X-Y data plot, histogram plot, and function plot. The "4051 Graphic System Operator's Manual" contains all the information that's on the tape, with amplification, plus an introduction, sections on "keys, buttons and switches" and routine maintenance, appendixes on error messages, specifications, and installation, a glossary, and an index. The fourth manual is the "4051 Graphic System Reference Manual."

System Software Tape

For the beginner, the first of the seven programs to use would be the Graphics System Tutorial, which has a menu with these six alternatives: the whole tutorial, keyboard operations, demonstration of graphic software, programming primer, graphics commands, index (for picking and choosing).

System verification, which is meant to be used when the system is first received, and "at any later time when system performance is in doubt," takes less than 10 minutes, and is mostly automatic. The Software Verification part requires the user to press keys as requested, and the system responds with various phrases and test patterns. The Firmwave Verification checks out system memory, and runs all by itself in less than a minute.

The two best features about all the canned plotting programs is that they're automated as much as possible, and there is a great variety of statements that let you do just about anything you can think of, plus some that might not have occurred to you. Let's look at a function plot to see how it all works. A function can be plotted in a very short time, if the user chooses to let the system set certain parameters automatically. If he/she wishes, the user can override these "default values" and, for instance, move either axis and/or change its length, change the data range on either axis, choose either a line or point plot, and choose from five point-plot symbols.

After the user has chosen Function Plot from the master menu, he/she is asked to enter the function, either single-variable in Y or double-variable in X and Y. The screen then shows the Function Plot menu, listing all the parameters one can enter. Most of these can be skipped by letting the 4051 set most of them with default values, but there are of course several that must be entered. This is done by selecting the first menu item, Display Function, which will then cause the 4051 to ask the user to enter only three numbers: beginning and ending independent variables, and increment.

After the increment is entered, and the user presses Return, the screen displays the function, complete with axes and labels. If the plot is satisfactory, that's it. But if the user wants to smooth out the curve, he can enter a smaller increment and replot in a few seconds. By calling up the Function Plot menu, he can make further changes as desired. If he forgets what changes he's made, selecting List Parameters from the menu will provide a reference chart.

As a BASIC computer, the 4051 is excellent. All the necessary statements are here, plus some I'd never heard of before which proved not only fascinating but useful. FUZZ decides just how close a comparison is to be made between two non-zero numbers; FUZZ 10, for instance, compares two numbers to ten digits. This gets around the problem of what are often necessarily imprecise mathematical operations. SECRET permits a program to be executed only; "it can never be listed, saved, or in general output from memory." SUM "returns the algebraic sum of the elements in a specified array."

Graphics

Now to the specialty of the 4051, graphics. With its amazing variety of statements, the 4051 can create just about anything you've got in mind, as far as static display goes. VIEWPORT and WINDOW determine which part of a curve will be shown where on the screen. AXIS produces an X-Y axis, with tic marks as desired. After you specify an initial point with MOVE and a pair of coordinates to indicate how far from the lower-left corner of the screen you want the point to be, a DRAW statement with coordinates will specify a line. A square can thus be drawn by using one MOVE and four DRAW statements. This can be simplified with arrays to a single DRAW statement by using DIM, four pairs of coordinates in DATA, READ X,Y and DRAW X,Y.

RMOVE and RDRAW "free the programmer from having to figure out the absolute coordinates of each data point" by interpreting the numeric constants as relative increments to the position of the cursor. ROTATE will move a pattern, or single vector, through a specified angle, in either direction, once or more than once (with FOR/NEXT), and is a main ingredient in creating one type of graphic art.

All this information on graphics is given in the Reference Manual, which also has sections on language elements (constants, variables, operators, strings, DIM, LET), environmental control (presets for degree/radian/grad, trace/normal, initializing the system, fuzzy comparisons, etc.), system control (CALL, COPY, HOME, PAGE), memory management (DELETE, MEMORY, SPACE), controlling program flow (END, FOR/NEXT, GOSUB/RETURN, GOTO, IF/THEN, RETURN, RUN, STOP), handling interrupts, input/output operations (DATA, OLD, PRINT, READ, SAVE, SECRET, etc.), math operations (ABS, COS, INT, LOG, PI, SQR, etc.), character strings (ASC, DIM, INPUT, LEN, VAL, etc.), programming editing, debugging and documentation (DELETE, LIST, REMARK, RENUMBER, SET), language syntax (rules for line numbers, keyboards, data items, etc.), and appendixes on error messages, tables (ASCII character values, character priority, fonts, etc.), interfacing information, and glossary.

The two manuals are very good, except that the examples in the reference-manual sections on the various statements are too skimpy in many places.

There's a lot more to say about the 4051, but if you're interested in this computer, a demonstration is of course much better than any number of words. Tektronix has several fine demo tapes, such as the one I had, which gives examples of applications in mechanical engineering (conic analysis), electrical engineering (filter design), business (financial analysis, depreciation, savings and loan), mathematics (integration), statistics (regression analysis), plus sections on peripherals, available character sets, stock-market bar chart, Crosshair Simulation (a sort of draw-it-yourself), three-dimensional plot, winding up with a Capability Demonstration, which describes the 4051, shows a wide variety of graphs, and ends by drawing the Tektronix logo.

If there's anything Tektronix left out, I wasn't able to discover it in four weeks of using this marvelous machine, the 4051 Graphics System.

A Creative Computing Equipment Profile . . .

HEWLETT PACKARD HP-25 CALCULATOR

"The Minimum Computer"

by James Blodgett

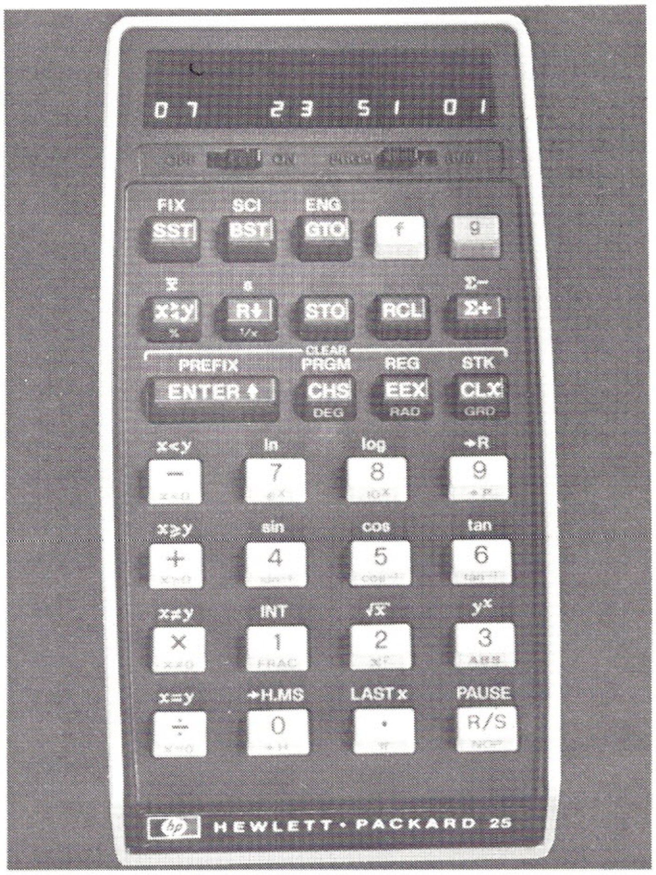
The Hewlett-Packard HP-25 programmable calculator is the cheapest complete computer system on the market today. This statement may raise debate. For one thing, I am saying that the HP-25 is a computer while several cheaper programmable calculators are not. Second, despite its simplicity I am saying that it is a complete system. Both points are matters of definition and debate, but my definition makes good sense.

What, after all, makes a computer? At what level of complexity does a programmable calculator become a computer? At present common usage gives no precise distinction. But since loops, nested loops, and alternate sequences of operations are fundamental to programming, I would like to suggest that a system is a computer if it can do them, and is not otherwise. In general, this means that it must be able to do some form of conditional branching. Besides, I have tried programmable calculators without conditional branching, and they are simply not much fun. They can do only one short sequence of steps in one way, so the complexity of their programs is strictly limited. Conditional branching allows complexity. And the HP-25 is the cheapest programmable with conditional branching.

(This is not to say that a cheaper programmable might not be the best buy for many people. For some practical applications, a little programming goes a long way. And on some models, it is possible to simulate simple loops by pushing some version of the "run" button over and over.)

Another possible point of contention is my calling the HP-25 a complete system. I call it complete because it computes, by itself, and it is all you have to buy. At this writing (Feb '76) there are more ambitious computers advertised in the price range of the HP-25, but they are by no means all you have to buy. For example, the Altair 680 CPU board in kit form lists for the same price as the HP-25 at retail ($195), although the 680 is not yet discounted while the HP-25 is. But the 680 CPU board is a "complete computer" only to those who are ready to wire it into something else—and that something else had better include some memory. Even the 680 "complete" kit with memory, front panel and case for $345 is a complete computer only to those who are willing to enter and read out both data and machine language instructions in binary, do this each time the machine is turned on, and flip switches forever. This might provide interesting recreational computing for some people, but the machine by itself is certainly not very practical. Most Altair users buy at least a terminal, the Basic language (available as of Feb. '76 only on the Altair 8800) and the extra memory necessary to support Basic, and these additions put the system in the $2,000 range.

One might also question my statement that the HP-25 is a complete computer because of features which it lacks, or because of the general limits of the system. Hewlett-Packard, for example, calls the HP-25 only a "programmable calculator," while their HP-65 is "fully programmable" because it can store its programs on small magnetic cards. Such storage is certainly a useful feature, but it is not a necessary feature for programming to intellectually "feel like" computer programming. The HP-25 feels like a computer, and it doesn't need a card to do so.



(The new Texas Instruments SR-52 also has a card, and twice the memory and program capacity of the HP-65 for half the price. Readers contemplating purchase might well consider the SR-52; the HP-25 is the cheapest but not necessarily the best choice.)

The HP-25 is in fact quite limited for a computer: it has 49 steps of programming and 8 memory registers, plus 5 registers in its operational stack. Refusing to call it a computer because of its limits, however, is mainly a matter of preference for larger systems. I would call the HP-25 a computer despite its limits. The system it defines is not at all trivial. It can do enough so that it is fun to program, and it is also useful, and will do a reasonable fraction of the personal computing that a mathematically oriented user would do with a much larger system.

Indeed, the very fact that the system is limited has both practical and recreational advantages. A larger system can be very impractical when one is seduced into wasting days solving what had at first appeared to be a simple problem, writing hundreds of steps, debugging, and improving the output. If one has an HP-25 and a problem can't be solved on it, one knows that the problem will take a substantial amount of programming on a larger system, and this is a very good point at which to ask whether the solution is worth the trouble.

Another recreational and educational advantage of a limited system is that one is much more quickly forced to optimize programs. Optimal programs are much better esthetically than sloppy programs, and there is a real feeling of craftsmanship in knowing a system well, trying every trick in the book, and finally being able to shoehorn a complicated problem into a limited number of steps.

An example of a recreational problem may give some idea of the possibilities and limits of the HP-25. I have been trying since I bought the machine to make it play ticktacktoe. I am almost certain that this is impossible in such a limited machine, but it is less impossible than one might think at first.

For example, there are nine positions in a ticktacktoe board—how can one store the contents of these positions in only eight memory registers? Well, the machine can take a number with a decimal point and throw away either the fractional or the integral component. Thus it is possible to extract a specific digit in a larger number by placing the decimal point in front of the digit, taking the fractional component of the number, then multiplying this fractional component by ten and looking at the integral component. It is also possible to place a digit in any decimal position by multiplying the digit by ten-to-the-power of the desired decimal position and adding the resulting number to the number in a storage register. (Unfortunately the ten-to-the-power function is slightly off in the last decimal place for ten-to-the-seventh and higher powers, but if necessary this can be corrected by a few steps of programming.) Since one can both take out and also put a number in any decimal position, the ten decimal places in one register can be used as ten different memory locations. Thus if necessary a representation of the ticktacktoe board can be stored in only one register.

A more fundamental limitation for implementing ticktacktoe is that the small number of branches which are possible in 49 steps seems much lower than the number of decisions a computer must make in playing ticktacktoe. The number of branches possible for the HP-25 is much less than 49 because a branch requires from one to five steps depending on what one is doing. In order to branch it may be necessary to bring the two numbers to be compared into the two appropriate registers, specify the logical test to be applied, and specify the two different directions to go to depending on the outcome of the test, and this totals five steps.

My latest thought is to sequentially peel off digits from a sequence of memory registers and use these digits as a sort of higher-level programming language to specify the sequence of application of a group of subroutines, thus in a sense expanding the programming capacity. I am fairly sure that this will not work, since the peeling-off and branching program alone will take quite a few steps, and there might be room for only about four or five very short subroutines.

Whether or not it can be implemented, the ticktacktoe problem is an example of the complexity and richness of strategy possible in what seems at first to be a very limited system. And despite the difficulty with ticktacktoe, by no means are all games excluded. For example, the *Applications Programs* book that comes with the HP-25 includes a moon-landing program and a version of nim. And I have written an ESP testing game, a ball-bouncing game, and others. Games which can be implemented tend to be relatively simple, however. The best game is the system itself.

Hints on Buying a Used Teletype

by David Ahl

If you've been following the Teletype ads, you know that the ASR 33 has been sold for well over 10 years. It was, for many years, the workhorse of the TWX and Telex networks (although the real heavy-duty workhorses of the networks are the Models 28, 35, and 37).

The ASR-33 is Teletype Corporation's most popular product in history with over 600,000 delivered as of early 1976. Many of these are now on the surplus market available both "as is" and reconditioned. Here are some hints if you're considering buying a used unit.

Serial Numbers 1-200,000 are likely to be dogs. In "as is" condition they're worth $250 or less. They're generally tough to refurbish because so many parts must be replaced. (By the way, the Serial Number is hard to find—you have to take off the cover and look for a little plate, generally covered with dirt and oil, in the back corner).

Serial Numbers 200,000-450,000 may be OK as is, particularly if they've been under a regular maintenance contract. As is price should be $350-$400 or so.

Serial Numbers 450,000 and higher should be in good shape. Nevertheless, it's worth seeking out units that have been under a maintenance contract.

Reconditioning generally adds $250 or more to the price; generally a reconditioned unit of any age is going to cost $700-$900. While this is only $100-$300 under the new price ($969 plus), sometimes a reconditioned unit is actually better — all the initial bugs have been worked out and it should be in good adjustment.

Of course, another alternative is to buy a new one. Base price direct from Teletype Corp. is $969. However, for most minis and/or micro systems, you're going to have to add about $75-$200 worth of bits and pieces from the computer vendor to make it all work.

The other big question—where do you get one? Many, perhaps most, second-hand computer or terminal dealers will not sell you an ASR-33. The reason, as a salesman from American Used Computer told me is that dealers can make a lot more money renting them than selling outright. Hence, you're going to have to make a few phone calls to terminal vendors listed in the Yellow pages or from ads in Computerworld, Computer Hot Line, etc. Chances are you'll find someone with a temporary overstock and maybe even someone who cares about hobbyists or schools and will sell you a '33. After six phone calls in the metro New York area I found 3 dealers willing to sell me a '33 for prices between $775 used and cleaned to $875 reconditioned. I eventually decided to wait out the 8 to 12-week delivery cycle and get one for $969 through the MITS-Teletype Corp. deal. In theory, delivery today is better, but don't bank on it.

There are many models of the '33, but by far the most common model in use by schools and hobbyists is the ASR (Automatic Send Receive) i.e., it has paper tape reader and punch, friction feed (as opposed to sprocket or pin feed on the paper). The order number is: 33 ASR/TC or 3320/3JA.

A Creative Computing Equipment Profile . . .

ODYSSEY VIDEO GAMES

by David H. Ahl

Ah, the fascinating path of product development. The original Odyssey game from Magnavox (1972) didn't beep. It didn't even keep score. It was expensive (in the multi-hundred dollar range). It came with overlays to put on the face of the TV set and little program cards which allowed you to play a fair number of games — skiing, submarine chase, shooting gallery, and, of course, tennis/ping pong.

However, it wasn't nearly as good as Atari's Pong (the original one to appear in taverns, arcades, and shopping centers). Magnavox responded by filing a suit against Atari claiming that they (Magnavox) held the patents on ping-pong/tennis ball bouncing algorithms for video displays. Panic in Atari-land until it was determined (with the aid of yours truly) that CRT ball-bouncing algorithms had been around since 1957 at places like MIT and CMU and other homes of computer hackers. Having won their case, Atari added insult to injury by putting together a home version of Pong, lining up Sears to distribute it, and pow(!) the battle for the home video game market was off and running.

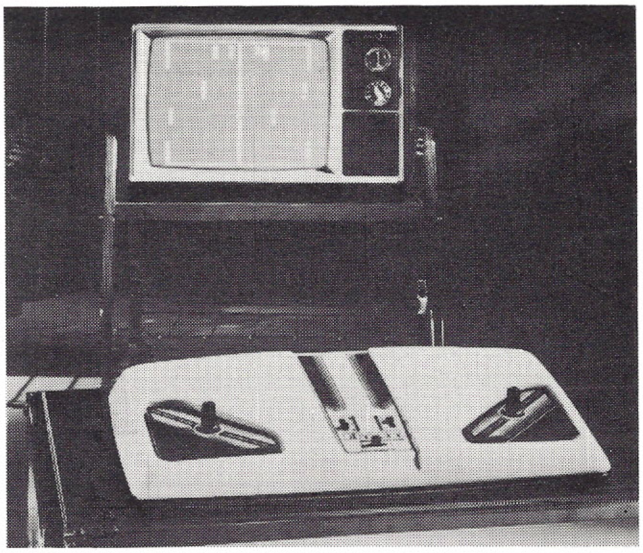
Magnavox, not to be outdone, introduced the Odyssey 200. In all fairness, I should point out that this product would have been introduced whether or not the suit was won. The 200 was, in most ways, an improvement over the first Odyssey. No overlays, no program cards, and it kept score and beeped. However, it had only three games — tennis, hockey, and "smash."

Tennis is the standard CRT tennis. Hockey has a wall on each side of the screen with an opening in it. The puck bounces on the wall; the object is to get it in the opening (goal). Smash is a wall ball game. There can be only two players, although in tennis and hockey each player can control a main paddle and a second, drone paddle (in tennis it's a doubles partner, in hockey it's a goalie).

A significant improvement over earlier TV games is the horizontal as well as vertical paddle control as well as ball control. Once you hit the ball, you can control its vertical path until it is hit by the opposing player. Sports purists will object that this is not an accurate simulation of the real thing, however, it makes for a much more interesting game. And why shouldn't electronic games be different, or even unique (heaven forbid)? There is also a ball speed control.

Because each player has three controls and only two hands, (at least those from this planet), the game can be very exciting or very frustrating. Since there are essentially two variables and one constant per player, there are many possible playing strategies. It is conceivably possible to have two or three players per side. This is not particularly satisfactory for casual play although things improve after a few beers and/or as team-mates get to know each other's moves.

This year there are three Odyssey games. The 300 is a basic unit with Hockey, Tennis and Smash. It has only one control and drones (goalies) only in hockey. The main additions this year seems to be an automatic serve and 3-position skill switch. It keeps score, beeps, and has automatic English. Retail price is around $69.



Odyssey 400 is essentially an updated 200 as described above with mainly an addition of automatic serve. Retail price is around $100. However, if you don't need automatic serve, you might want to look for a 200; in this area they're being heavily discounted to make room for the newer 400.

Odyssey 500 is the top-of-the-line model with colored playing fields and players. It also has a fourth game, soccer.

Magnavox loaned us an Odyssey 300 for testing. It seemed rugged and, as they say in *Road and Track*, the controls fell nicely to hand. It connected to the VHF antenna terminals easily. The only pain was finding 6 "C" batteries to make it work (my DC power supply was busy elsewhere). Since the unit was a loaner I didn't explore the insides too extensively to see if any interesting mods were possible. I wouldn't encourage this type of thing anyway; it doesn't use a microprocessor or memory so it wouldn't have much general purpose versatility. (Please don't take that statement as a challenge to prove me wrong.)

Summary: if you like eye-hand coordination games and have $70 (or more) burning a hole in your wallet, Odyssey might be opere pretium.

If you're a comparison shopper, you should be aware that Atari/Pong now have four games on the market this year starting with the bottom-of-the-line PONG IV ($65) and ranging to a 4-player, 5-game super model ($100) available through Sears. J.C. Penney has still another model ($60). Unisonic's Tournament 2000 plays 6 games and includes a separate electronic pistol ($88 in K-Mart). Also several electronics parts vendors are selling kits in the definitely non-bargain price range of $50 - $90.

BUILDING A MITS ALTAIR 8800

GETTING A SYSTEM TOGETHER

Richard Kuzmack
Chesapeake Microcomputer Club, Inc.
1435 Layman Street
McLean, Virginia 22101

In this year's first issue of Creative, *Vol. 2, No. 1, we published Steve Gray's "Building a MITS Altair 8800: First Impressions." It is only fitting that in this last issue of the year we follow-up on that article with an account of interfacing-to and using the Altair 8800 once built. We asked Rich Kuzmack, President of the Chesapeake Microcomputer Club, if he'd write the follow-up for us, and this is his account.*

Once you have completed and checked out the assembly of your Altair 8800 main frame kit you are ready to develop a usable system by adding memory, input/output interfaces, and the input/output devices themselves. This is a good point to review your system plan, or start one if you have neglected this step thus far. Even if you are just beginning to think of maybe getting a personal computer of your own, stay tuned. In this article we will explore together the kinds of decisions you'll probably have to make in planning your system and the considerations you'll want to keep in mind as you make them.

The process of making a system plan usually consists of deciding how to get the most out of what you have the least. The money that you are able to devote to your system is most often going to be the limiting factor, although some will find the availability of time to work on it even more limiting. Neither is necessarily an absolute limit once you realize that you can substitute some patience for some of either one. It may take you longer to save up the money or assemble a board, but over the long haul you will have fewer regrets and headaches if you take the extra time required to avoid low performance components and hasty construction practices.

The other side of the system planning process makes you think about what you want to do with your computer. Indeed, there is a certain logic to planning what to put in to a system based on what you want out of it, but many computer hobbyists have not been especially concerned with this question. So let's not address the end use, but think instead in terms of the high level programming languages you'd like to have running on your system. Several are available for your Altair's 8080 CPU, and the amount of memory and I/O needed to support a particular language can provide you with a first cut at some goals for planning your system. A popular starting point that makes a lot of sense is the BASIC language because it can start small and grow with your system, from Tiny BASIC to the 4, 8, and 12K versions available from MITS and others. Even if you want to eventually be running FORTRAN or APL, one of the smaller BASICs is a good idea for a start. It will come in very handy later on for that large applications program that won't fit after one of the larger language interpreters soaks up its share of the available memory space.

Getting In . . . and out

Two kinds of input/output will be needed before your system is ready to run. First, some means to get to and from a non-volatile mass memory medium, such as audio cassette or paper tape, will be required to take advantage of available machine-readable software and to provide a way to save software you've written or modified yourself. Second, you will want a keyboard for program and data entry, and an output display of some kind. There are many possible solutions to performing these functions because of the many products on the market today that were designed to work in your Altair, and because of the variety of both new and used equipment originally intended for commerical systems.

In deciding what to get for my own system an important factor was the flexibility that could be squeezed out of each board. A board that can be made to serve more than one use saves both time and money that might otherwise go to buying and building another board and adding another slot (maybe even another mother board, too!). I made my decision and purchases several months ago and am quite pleased with my choices, but your circumstances and goals may be very different from mine and there is certainly more to select from now. While I hope the specifics of my system are helpful, you'll have to extract and modify to suit your own needs.

The input/output interfaces in my system consist of three boards: the MITS four port parallel board, the Processor Technology video display module (VDM-1), and the MITS audio cassette recorder board. The advantage of this configuration is that two of the three devices to be interfaced are already available in most homes: a black-and-white television set and a cassette recorder. Only the keyboard has to be bought, and one that is encoded for ASCII (American Standard Code for Information Interchange) can usually be had for under $50 on the surplus market, while keyboards with other encoding schemes cost a lot less but might have to be translated to ASCII in software. (It only takes 512 bytes to go both ways for an 8-bit code, but if you don't want to spare the memory, stick to an ASCII keyboard.)

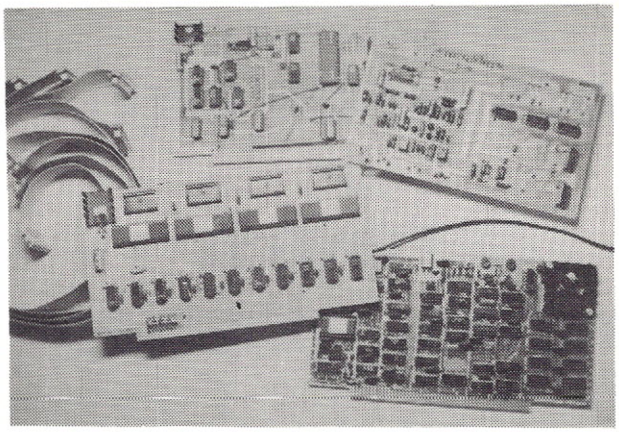
The parallel board was purchased initially with just one port to connect to a surplus keyboard. Using the Motorola 6820 Peripheral Interface Adapter (PIA) chip, it is an extremely simple, tremendously versatile piece of hardware. But there's a catch — in order to get the thing to work at all you have to configure the little beast with software that is more than a little complex. I thought it was my own stupidity that was the cause of my problems with it until I started getting phone calls from other members of our local computer club who were having the same kinds of trouble. The inadequacy of the documentation that came with the kit was eventually recognized by the presentation of expanded documentation in Computer Notes. Hopefully the explanation supplied with the kit has also been revised by now.

The Video Display Module presented the opposite problem. The instructions for using it were quite adequate, but the assembly manual is confusing in the great amount of detail concerning intermediate checks and tests, some requiring an oscilloscope which I don't have. So instead, I simply put the board together carefully without stopping for any of the intermediate checks, and winding up with a board that seemed to work fine. I attributed the difficulty I was having adjusting the picture to the TV set I was using until I found out that two diodes had been left out of the kit and the assembly manual. Installing the diodes solved the problem, and the current version of the VDM has corrected this ommission.

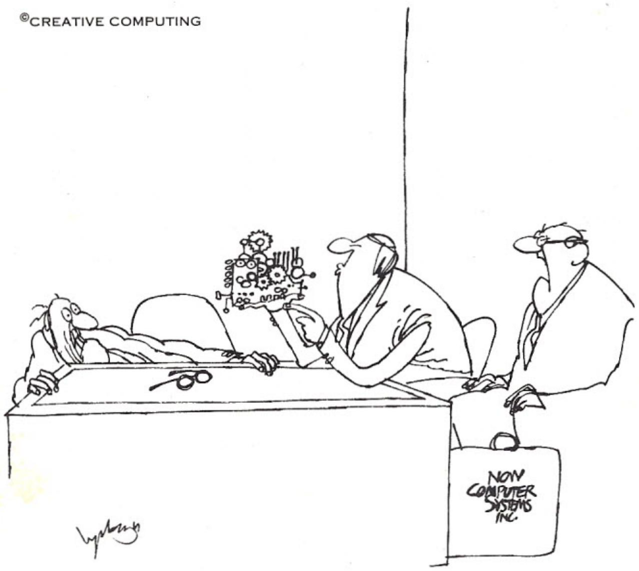
Modifying an old TV set to work with the VDM seemed beforehand like it would be a fairly formidable task, but following the guidance provided in Don Lancaster's *Byte* article on television interfacing (reprint provided with kit) proved to be quite straightforward. Getting the parts and the schematic for an old TV delayed things somewhat. The Sam's Photofact for *my* set was temporarily out of stock, as usual, and there is no point in even opening the set's cabinet until you've studied the circuit diagram and the component layout. The few parts that are needed were not available in the indicated type numbers at several parts sources, but after running a few poor stockmen ragged I was able to get equivalent substitutes for one of the three suggested circuits. A useful technique in such situations is to have a list of part numbers for each alternative made up ahead of time and when one of them can be filled completely note down beside each part the designation on the substitute part. That way when you get home you won't have trouble figuring out what was a substitute for what.

The audio cassette recorder interface consists of two separate circuit boards: a TTL-level serial I/O port and a modem (modulator-demodulator) board. Assembly of the boards presents no particular difficulties. The assembly instructions, however, say to mount the two of them together, but that way the unit will take up the space of two slots inside your Altair case. Then, too, I had another reason for not mounting them together. For audio cassette recording and playback the modem at that time operated on the same 2025/2225 Hertz band that a standard "103" style originate modem uses to receive data. Of course, if the modem will only be used with magnetic tape, just the 2025/2225 band is needed. However, I also wanted to use mine as a 103 data set so for this it had to be able to send on 1070/1270 Hertz and receive on 2025/2225 Hertz.

The ACR modem has the capability to do both jobs by changing the six jumper wires that control the modulator frequencies. To have both a standard 103 originate modem and an audio cassette recorder modem all in the same board (but not both at the same time) simply replace those six jumpers with a non-shorting six pole, two position rotary switch, connecting the numbered row to the "B" row for audio cassette operation and to the "A" row for use with an acoustic coupler to your telephone handset.



This dual capability of the modem board is no accident, by the way. It is used as a standard 103 style originate modem in the MITS COMTER 256 terminal. However, in order to make the modem less sensitive to the speed variations of low quality cassette recorders MITS announced a modem modification changing the width but fortunately not the center of the operating frequency band. That modification would make it much more difficult to use the same modem board for both applications. As I was having no trouble with speed variation on my cassette recorder I decided not to implement the modulator portion of the change, but only to widen the lock range on the phase locked loop of the demodulator. Thus I can read tapes written with either the old method or the new method, plus easily switch over to full 103 style operation. The only wrinkle that has been introduced into my plan by the modification is the potential problem that might come up if I wanted to give a tape written on my system to someone using a poor quality recorder, but even that would only require a brief loan of my recorder to get the data into the other system one time to then be dumped out using that other system's equipment alone.



"... you forgot to tell him it's an antique computer. He thinks it's ours..."

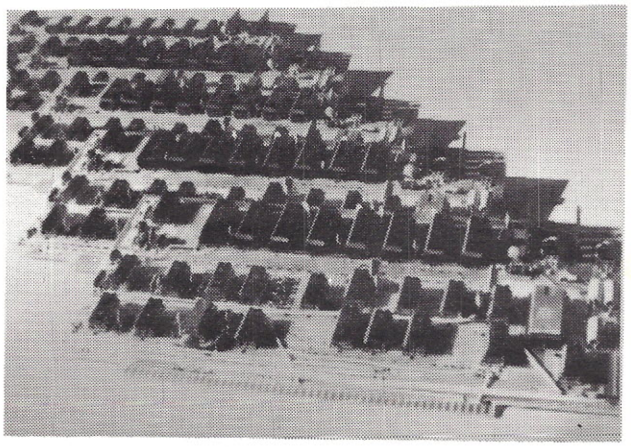
Deciding About Memory

Some of the toughest decisions you'll face with your system will have to do with internal system memory. You have more different memory boards to choose from than any other type of board. Most of this memory will be random access read/write (RAM), but at least some read-only memory (ROM) can save you a lot of work each time you turn the power on. Programmable ROM of the erasable variety is the best bet in the long run, because you can count on wanting to make changes from time to time. But it is RAM that requires the greatest care since it is needed in relatively large amounts.

Cost is certainly a most important factor, but do not neglect the relative advantages and disadvantages associated with the options you have in access speeds, board densities, and rates of power consumption. In some cases you may also want to give special consideration to some of the design features which may or may not be available on particular memory boards. These features include battery backup capability, wait state flexibility, switch selectable addressing, and memory protect features. When you get to the bottom line, however, you'll see that what really drives the decision process is your answer to the question, "How much memory do I want in my system when I'm all done?" You will also want to consider, "How will this memory board perform if I someday want to upgrade my ALTAIR from the 8080 CPU to some new processor chip?" (Selected chips of the Z-80, for example, can run at a clock rate of 4 MHZ, which means memories with access times longer than 250 nanoseconds would need a wait state to work in such a system.)

Well, there it is, again! We're back to the need for a good system plan, including a pretty solid idea of the use to be made of the system. But how else can you make an intelligent decision? If you'll need a lot of memory you had better plan to conserve power consumption, while a smaller system memory would allow you to use some of the really low cost but power-hungry memory boards. Then, too, memory can be added in increments of either 4, 8, or 16K bytes with readily available kits, and although the 4K boards may be easier to afford, the higher density boards may be a better buy in terms of cost and power consumption per byte. They will certainly take fewer slots for the same total memory size, and adding slots to your ALTAIR takes both time and money.

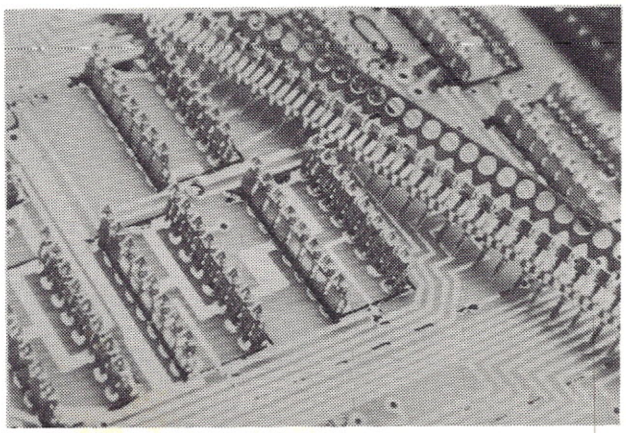
In my own system I decided in favor of high speed memory and started off with 20K of dynamic RAM in five 4K boards from MITS. There was a prolonged delay in getting them up and running because the boards had to be recalled for factory rework and replacement of defective parts. In the meantime MITS lowered its price on that kit and gave a substantial credit on a board-for-board basis for earlier purchases. So even though I could have done better with higher density boards now available based on the regular prices, the credit tilted the decision in favor of five more 4K boards.



Yes, that's a lot of memory, but my system plan includes applications needing that much. In particular, I have become very fond of the APL programming language, including an extensive library of programs that I've written over the past few years. When APL becomes available for the 8080 it is expected to require 24K bytes for the interpreter, on top of which memory will be needed for user workspace. Since APL has the greatest memory requirement of the uses I have for my system, it serves to determine the amount of memory I need.

Plan for Repairability, Too

As you implement your system plan you will be pouring in a lot of money and you can expect it to work for a long time before any of its parts fail. But eventually something's going to go bad, and it is now while you're putting it together that you should allow for the future event. My plan for repairability consists of never soldering an integrated circuit directly into a printed circuit board. I use sockets supplied with the kit and socket strips for all other ICs. If a board doesn't work because of a bad IC somewhere, it is then quite easy to pull and replace one IC at a time until the culprit is found.



I have used over 5,000 of these strip sockets and recommend them on three grounds in addition to the fact that I've never had any problems with them. First, they are relatively inexpensive when purchased in strips of 1,000, costing about a penny a pin. Second, they allow you to get at the printed circuit traces under the IC, which regular sockets don't. This feature saved me a lot of trouble when MITS announced that a trace had to be cut that was under the memory protect flip-flop on my 4K boards. Third, while they may not last through as many insertions and removals as a regular socket, if it becomes necessary to replace them it's easy because each is a separate piece to be unsoldered. Removing a regular socket would be as difficult as removing an IC.

Finally, while I have you thinking about repairs, you should make a simple addition to your ALTAIR that will lessen the likelihood of its needing repair. Add a metal oxide varistor (GE-750) across the power line at the terminal block to protect your system from damage due to high voltage transients. With hundreds or thousands of dollars of hardware inside your ALTAIR case, you can hardly afford not to invest the three dollars or so that it takes for this protection.

BEYOND BASIC

by Alan B. Salisbury

INTRODUCTION:

There are many levels of exposure to and interest in computers. It's probably safe to assume that the majority of Creative Computing's readers are familiar with BASIC and the computing power which it makes available to them. An equally safe assumption is that many readers do not have a very deep knowledge of the computer beyond that which they obtain through BASIC. This level of computer interest and understanding is sufficient for many purposes and represents a very large community of computer users.

The vast majority of users of computers have little or no knowledge at all of the computer. In fact, they may not even be aware at times that they are using a computer! These "users" include, for instance, the people who drop a quarter into a slot to play a tennis or hockey-type game using a video screen and hand controls. Slightly more involved, perhaps, is the person who sits at a terminal to play STAR TREK after someone else has loaded in the correct program and set the system up ready for play. Countless other applications could be listed in which the fact that a computer is doing the behind-the-scenes work may or may not be apparent to the user.

This article looks in the other direction, beyond BASIC, toward a deeper level of understanding of the computer. The purpose of twofold: first, it will help the reader to put BASIC in its proper perspective in relation to the many other types of computer software; second, it may (hopefully) stimulate many readers to expand their knowledge of computers and thus open up whole new areas of excitement and challenge.

BASIC IN PERSPECTIVE . . .

To begin with, let's review exactly what BASIC really is. Simply stated, BASIC is a programming "language." Like any language, it has a vocabulary with precise meanings and a set of rules as to how that vocabulary may be used (semantics and syntax). BASIC can be used to express problems and their solutions in a form that is readily understandable by humans—it looks reasonably close to English and algebra for instance.

Generally, BASIC is *not* directly understood by computers! (There are a *very few* special computers which actually are built to directly understand BASIC, however.) Hence, a translation is required from a user program written in BASIC into another form which is directly understandable by the computer. BASIC is therefore usually referred to as a "higher order language" (HOL) when compared to a "machine language."

There are other higher order languages, of course, in addition to BASIC. The most popular of these are FORTRAN (FORmula TRANslation), ALGOL (ALGorithmic Oriented Language) and COBOL (COmmon Business Oriented Language). A host of additional languages exist, highly tailored to specific uses. All of these languages are similar in that they require translation into a machine language before the computer can actually run the program.

COMPILERS . . .

Higher order languages wouldn't be of great value if the programmer was saddled with the job of doing the translation from BASIC, for instance, to machine language. Fortunately, this is the type of job that a computer can do very well. Programs called "compilers" have been written to do this translation for us. A compiler takes as its "input" a higher order language program (called the "source program") and produces as its "output" a machine language program (called the "object" program) ready to be run on the computer. We thus have a two-step process including a compile (translate) phase and an "execute" (run) phase.

One of the important and very nice features of higher order languages is that they are generally "machine independent." That is, when writing in BASIC, the programmer doesn't need to know if his program will be run on an Interdata 7/16 or a DEC PDP-11/40 for example.[1] Different translators will, of course, be required, one for the Interdata machine and one for the DEC machine, since their machine languages are quite different. These translation programs are usually provided by the manufacturer of the hardware.

MACHINE LANGUAGE . . .

We have mentioned machine language many times without really addressing how it differs from higher order languages. This can best be understood by looking at the familiar four-function calculator, which in many ways is a very simple computer. This calculator can do four functions: add, subtract, multiply and divide. This is, in effect, the vocabulary of "instructions" which the calculator understands. We can use the four-function calculator to solve any problem whose solution can be reduced to a series of steps (a "program") using the +, -, X, ÷ operations.

As an example, let's consider the BASIC statement:

```
LET X = A+5
```

We can translate this into a series of steps (using the calculator) such as the following:[2]

```
Clear
Enter A
Depress +
Enter 5
Depress +
Read X
```

Here we have expanded the vocabulary to include the human actions (clear, enter, depress for input, and read for output). In the language of the machine/human team, the above program is a translation of the BASIC statement.

Moving from the four-function calculator to a minicomputer is not difficult. One major difference is that the

[1] In fact, some differences in BASIC may occur between manufacturers.

[2] The exact sequence will depend on the logic of the particular calculator. The sequence shown is typical.

vocabulary, or "instruction set," is greatly increased; a typical mini may have 70 or more functional instructions which it understands, including the familiar add, subtract, multiply and divide. New capabilities here include such things as "shifting" numbers left or right, or "comparing" two numbers to determine if one is bigger, smaller, or the same as the other.

Another significant difference is in the amount of memory available. Our calculator example, above, did not include memory. Many calculators have one or more "memory" locations in which a number may be "stored" and later "recalled" by use of appropriate keys (which actually add to the instruction set vocabulary of the calculator.) Computers generally have *thousands* of such memory locations which can be used to store not only numbers (or "data"), but they can also be used to store instructions so that the entire program can be stored within the computer.

It is this last fact (program storage) which makes the computer the powerful instrument it is. With the program stored inside the computer, the operator need only depress the "start" or "run" switch and the program can then be executed at electronic speeds, typically in *millions* of instructions per second. Add to this the fact that the computer can make "decisions" (for example, using a "compare" instruction, go on to *different* parts of the program depending on the results of the comparison) and you have a good explanation of the power of the computer.

Returning to our previous example, a sequence of minicomputer type instructions to perform the LET X = A+5 function might be

```
CLA  A
ADD  Five
STO  X
```

CLA, ADD and STO are abbreviations (called "mnemonics") for the full instruction names. For example, CLA could be "Clear and Add," meaning "clear the working (accumulator) register to zero and add to it the contents of the memory location indicated" (in this case, A). ADD would simply be "add to the accumulator register" without clearing beforehand, and STO would be "store" the contents of the working register in the indicated memory location. Each of the many instructions of the computer has a similar detailed and precise meaning.

BINARY NUMBERS

In the binary number system, only two symbols (digits) may be used, 0 (zero) or 1 (one). The decimal system, on the other hand has ten symbols: 0 thru 9. Just as in the decimal system, when counting causes us to run out of symbols in one column, we carry 1 into the next column and start over, the same process occurs in the binary system. The difference is that a carry in the decimal system occurs for every count of ten, while it occurs in the binary system for every count of two. Therefore, "place values" in the binary system are powers of 2 (1, 2, 4, 8, 16, etc.) as compared to place values in the decimal system which are powers of 10 (1, 10, 100, 1000, etc.).

Conversion from binary to decimal is a simple process requiring only that the place values corresponding to 1's in the binary number be added up. As an example:

```
                                      32
64's 32's 16's 8's 4's 2's 1's         8
                                       4
 0    1    0    1   1   0   1   =      1
                                      ---
                                      45
```

To complete the picture, we have to now point out that even the above three instruction program sequence is not really in *machine* language. Computers don't normally understand letters and words such as CLA. In fact, everything within the machine must ultimately be in the form of binary numbers, consisting only of combinations of ones and zeros. This is the fundamental unit of information within the machine, known as the "bit" (for binary digit).

Memory locations and working registers within a computer generally contain a fixed number of bits which is the "word" length of the computer. Minicomputers typically use 12 or 16 bits per word; alternatively, as few as 8 bits (the "byte") or as many as 64 bits may be handled at a time. The significance of word length lies primarily in the magnitude of the numbers that can be stored in a single word; fewer bits means smaller numbers or, more important, greater round off error since, in effect, fewer significant digits can be saved. Speed is also affected, since a wider word length usually means that more bits can be processed at one time.

It was mentioned earlier that memory is used to store both data and the program. Program instructions are stored in memory words according to precise formats. As an example, a 16-bit word may be divided into 6 bits for an "operation code" (specifying the particular instruction to be performed) and 10 bits for a memory address (the location of the data to be used). Our CLA A instruction would thus include a 6-bit code for CLA (it might be 001010) and a 10-bit binary address for the location we chose to call A (this could be 0000101101 if location 45 were used for A).[3] The complete machine language instruction would then be:

```
001010  0000101101
Opcode   Address
```

ASSEMBLERS . . .

Once again we are faced with the problem of translating from the mnemonic (or symbolic) form of instruction into the numeric machine language form. The computer comes to the rescue as before with a special translation program (available from the hardware manufacturer) which "assembles" machine language programs from the symbolic instructions. An "assembler" program takes as input symbolic assembly language source statements and produces as its output a binary machine language object program ready for execution. Unlike higher order language programs which are largely machine independent, assembly language programs are very machine dependent since each type computer has its own, generally unique, machine and assembly language.

In addition to relative machine dependence or independence, there are other considerations involved in understanding the differences between compilers and assemblers. Our example showed that a single BASIC statement resulted in *several* assembly language-type instructions. Each assembly language instruction, on the other hand, usually causes a *single* binary machine language instruction to be generated. It is not uncommon for a good assembly language programmer to write a program requiring fewer machine instructions than one written in a higher order language and compiled into machine instructions. The assembly language programmer may therefore find that his program requires less memory and executes faster. The price paid for this possible bonus is usually the extra time (and training) required to do assembly language programming. As compilers become more efficient, this difference may narrow.

[3] See inset box for an explanation of binary numbers.

LOADERS ...

We have talked about compilers, assemblers, source programs, and object programs. How do these programs get into the machine? One (horrible) alternative is for an operator to load the binary instructions through the console of the machine, one word at a time by setting switches or entering them through a small calculator-like keyboard. Preferably, the program could be punched onto paper tape or cards beforehand and then simply read into the machine. The capability to "read" a program into memory requires that another program already must be in memory to cause the reading to take place! Such a program, called a "loader," must include all of the many detailed instructions required to read a card, for instance, and then move all of the 80 characters or other binary information read from the card into designated locations in memory. In fact, all input and output operations are fairly complicated at the machine language level and the manufacturer usually provides "utility" programs so that the programmer doesn't have to write these himself. For the same reason, a "library" of other useful routines is generally available including such things as math functions.[4]

SYSTEMS SOFTWARE ...

Compilers, assemblers, loaders and utilities are collectively referred to as "systems software" and are usually written by the manufacturer's "systems programmers." The using programmer is, in contrast, referred to as an "applications programmer" and his "applications programs" are the programs written to solve the user's problem.

Starting with a program written in BASIC, there are many steps required before the output is available. First, a loader must be in the machine. (Loading the loader is itself a job to be done by the operator!) Then the BASIC compiler can be loaded in, the source program read, and the translated object program produced as output. Now the object program is read in (again by a loader) and then executed producing the final results. Input/output utilities and math routines may well have been required and loaded in along with the object deck.

The smallest mini (or micro) computers are often operated in just this fashion, with the operator handling many separate programs and manually controlling the sequences of loading and execution. This job, too, is one in which the computer can lend a hand.

OPERATING SYSTEMS ...

The most important of "systems" programs is one called an "operating system" (sometimes "executive," "monitor," "control program") which takes over these tasks of scheduling, allocating space in memory, calling other systems programs, etc. With an operating system (OS), the using programmer can simply state the kind of job he wants to do (using a special "job control language" requiring a card or statement of the beginning of his program), and the OS will handle many of these details for him. Here, then, we have another form of computer language, that which is used to talk with the operating system.

With an OS in control of the computer, our sequence of events is simplified. The source program deck, written in BASIC, together with a job control card, may be all that the user must place in the input card reader. After reading the job control card, the OS will call in the BASIC compiler (probably stored on a disc file), load it, translate the source program to a machine language program, load the machine language program together with all required utility and library programs, and finally execute the object program. What we have described here can be called a "Compile-and-Go" scheme, in which the system does not have to stop between the translation and execution phases.

INTERPRETERS ...

Our brief tour of systems software has omitted one important type of program, the "interpreter," which is particularly important to BASIC users. The use of a BASIC compiler to first translate a BASIC program into machine language for later execution is one method available for the BASIC user to execute his programs. It may be the only method, depending on the computer and its available systems software. Frequently, however, a BASIC interpreter may be available to execute BASIC programs, either in addition to a compiler or in place of it.

An interpreter differs from a compiler in one major area: an interpreter does not *translate* the source program, but rather it effectively *executes* it directly. Each line of a BASIC program is examined by the interpreter to determine what actions will be required to execute (or, more accurately, evaluate) the statement. The interpreter then performs those actions by immediately executing the appropriate portions of its *own* program, and then moves on to the next source BASIC statement. This is in contrast to the compiler process in which the entire BASIC program is translated to a unique set of machine language instructions which are then executed after the compiler has finished its job.

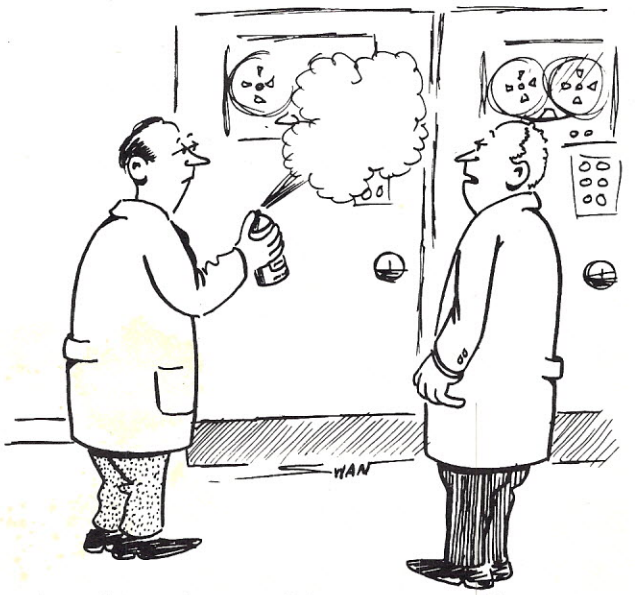
To illustrate how the two systems differ, let's consider how they would process the BASIC statement:

```
LET X = 2+3
```

A compiler would produce as its output a machine language *program* consisting of several instructions, which *when executed later* would calculate the desired result and assign it to the variable X. An interpreter, on the other hand, would produce as its output the new *value* of X which is 5; no instructions or new program would be produced by the interpreter.

What differences does the user see between a compiler and an interpreter? Not many. Both methods ultimately produce the same end results. If the program is to be executed many times, there is a benefit to using the compiler method; in this method, it is possible to obtain a copy of the object program produced by the compiler, on punched cards or tape for instance. Then when the user

©CREATIVE COMPUTING



"Where did you learn to debug a program, Haverstraw?"

[4] Hardware math instructions are normally limited to add, subtract, multiply, and divide. Trig functions, square roots, etc., require programs that compute these higher math functions using the basic instructions.

wants to execute his program again, he can merely load it and execute it, without going through the time consuming translation phase. Compilation therefore offers the potential of time-saving efficiency.

INTERACTIVE SYSTEMS ...

Interpreters generally offer an advantage of their own in that they may be "interactive." This allows the programmer to sit at a teletype, for instance, with the interactive BASIC interpreter in control, and get immediate "feedback" from the interpreter as he enters his BASIC statements. Errors can be immediately identified by the interpreter and corrected by the programmer. Also, the results of executing each BASIC statement can be available immediately, thus making the system react to the user as if it were a sophisticated calculator. For a new program, it may therefore be possible to get an answer faster using an interpreter than using a compiler, since the interpreter does not have to go through the separate translate and execute phases.

DEBUG PROGRAMS ...

A special type of interactive program often available for mini's and micro's is the debug program, sometimes called a monitor.[5] A debug program is used to aid in debugging a machine language program. It typically permits the programmer to insert "break points" in his program which will cause the program to pause in its execution and then allow the programmer to examine the contents of key registers and memory locations. In this way, the programmer can walk through the execution of his program and if there is a "bug" in the program, he can isolate it (hopefully).

TEXT EDITORS ...

Other programs besides interpreters may be interactive. One of the most common is a "text editor" which can be used to help create a new program or "file." If a BASIC compiler were available through a time-sharing system, an interactive text editor could be used to build or create a BASIC program using an on-line terminal; the text editor would permit the programmer to make changes such as adding or deleting individual characters or whole lines. When the programmer is satisfied with his program, he could then ask the system to compile it and execute it. The software performing the actual time-sharing operations serving many users simultaneously is essentially a more complex type of operating system as we have previously discussed.

SUMMARY ...

We have now completed a very general introductory overview of systems software. The reader should now have a reasonable understanding of the various types of programs involved, the functions they perform, and how they relate to one another. The types of systems programs available and the functions they perform are summarized in the accompanying table.

[5] This use of the term "monitor" is somewhat different from the OS type "monitor."

TABLE OF KEY SYSTEMS PROGRAMS

Systems Program				
Name	Function	Input(s)	Output(s)	Comments
Compiler	Translate HOL Program to Machine Language Program	HOL Source Program	Listing. Machine Language Object Program	Object Program may be punched out or loaded into memory
Assembler	Translates Assembly Language Program to Machine Language Program	Assembly Language Source Program	Listing. Machine Language Object Program	Object Program may be punched out or loaded into memory
Interpreter	"Executes" HOL Program	HOL Source Program	Problem Solution	Output is only what the source program produces. Interactive
Loader	Loads Machine Language Programs	Machine Language Object Program(s)	Machine Language programs ready to be executed	Loaded into memory
Text Editor	Creates or edits files	Source Programs, data, etc.	Listing or copies of files when requested	Interactive
Debug Program	Facilitates isolation of program bugs	Commands to Debug Program	Responses to commands, e.g. register contents, memory contents, etc.	Runs along with user's object program to be debugged
Operating System	Overall control of computer system and its resources	Job Control Language or equivalent. All other inputs to computer	Responses to JCL commands. Log of system status, error messages to operator, etc.	Schedules jobs, allocates memory, etc., to minimize need for operator

SNOBOL

by David Touretzky

SNOBOL is a string processing, pattern matching language, that breaks all the restrictions associated with numerical languages like FORTRAN or dp languages like COBOL. Here are some key points:

1. No fixed data types. A variable's type is determined by its values. Changing the value changes the type. The datatypes available are: string, integer, real, double precision, array, table, pattern, name, and compiled code. The table is a type of hash table: its subscript is a string instead of an integer.

2. The heart of SNOBOL is the concept of pattern matching. A SNOBOL statement is made up of one or more of the following fields:

```
   label  string  pattern  =
      string  branch
```

Every pattern match either fails or succeeds. The branch section causes branching to a specified label based on the results of the pattern match. For example:

```
   MYLABLE  CARD  'CAT'  |  'DOG'  =
      'ANIMAL'   :S(L1)
```

In statement MYLABEL, the variable CARD is searched for any occurrence of 'CAT' or 'DOG'. If the test succeeds, the string which was matched is replaced by the string 'ANIMAL', and, since the match succeeded, the program would branch to L1. If the test failed, the program would continue with the next statement. Pattern matching can be much more complex, and can include many levels of alternatives, calls to user or system functions, recursive pattern definitions, and self-modifying patterns. Many special characters are used in pattern matching operations. For example, the $ is an assignment operator.

```
   TEXT SPAN('A')  $   C  =
```

will match the first contiguous string of A's in TEXT and assign that string to the variable C. In addition, the string will be deleted from TEXT, since a null expression appears to the right of the equals sign.

3. Concatenation is accomplished by writing expressions next to each other. For example:

```
   MESSAGE  =  'THE BILL IS'
      (COST  ★  1.05)  '.'
```

In this case, COST would be multiplied by 1.05, converted to a string, and concatenated with the other strings in the expression. Strings may be of any length. They expand and contract through pattern matches or assignments.

4. Complete debugging tools are available, including traces, variable dumps, and program controlled error handling. Interactive implementations, such as SITBOL, allow convenient debugging as the program executes.

5. As you have seen, SNOBOL has no verbs, no data restrictions, and no artificial constructs like blocks, procedures, or cases. The language is based on special symbols, such as =$.★?&:()@, and system functions which generate patterns (such as CONVERT). There are also some system variables, such as &ALPHABET which contains every possible character, &STLIMIT, the maximum number of statements that may be executed and &ERRNO, the code number of the most recent error interrupt.

6. SNOBOL is available on many machines, including the IBM 360, the DEC system 10, and UNIVAC 1108. Two extended versions are commonly in use: SPITBOL and SITBOL. The best introduction to the language I have seen is the *SNOBOL 4 Primer*, by Ralph and Madge Griswold, published by Prentice-Hall.

7. SNOBOL can be used for many things, including file manipulation, database editing, music generation, language analysis and translation, game theory, computer assisted instruction, and the simulation of automata.

"And where were you when the FORTRAN hit the fan?"

A RETAIL COMPUTER STORE? YOU GOTTA BE KIDDING!!

by Jim Dunion and Ron Roberts*

FAR FROM THE MADDENING CROWD

The doors are boarded up now, and the raging mob seems to have subsided. For awhile at least I'm safe. The light is flickering low on this, my last candle, and I realize that time grows short for me to put these thoughts onto paper. For tomorrow brings a new day, a new mob, a new call from David Ahl wondering where this article is. Time indeed, grows short. One year in the retail computer store business. One year. Is it possible that only a year ago this madness descended upon me? That only a year ago I was a reasonably happy, nonchalant student, quietly pursuing a graduate degree in computer science and then -ah- but now I'm getting ahead of myself. First of all, what we're talking about is retail computer stores in general, and the experiences of one such store, The Computer Systemcenter, in particular. What we have seen in this past year is no less than the birth of an industry. Now, with a full year under our belt, it is time for the first self-assessment. Retail computer stores — from where did they come, how have they developed, and where are they going? While we certainly can't speak for all stores, we can provide insight to the structure and events surrounding one such store's existence.

THE BEGINNINGS

In recent years, there has been an increasing tendency for technology to advance faster than society can adapt to its changes, a fact particularly visible within the field of solid state electronics. Most of those involved in either electronics or computer technology are aware of the recent and continuing advancements which have led to the development of the "computer on a chip." Although the implications of readily available low cost computer power have been discussed many times in many different forums, it has been almost invariably within technical, non-public circles. For the general public the usual interaction with any form of computer technology has been via the "DO NOT FOLD, STAPLE, OR MUTILATE" admonishment. To John Q. American, it has little mattered whether the phone company used their 1st, 3rd or 300th generation equipment to produce his bill. To him, a bill is a bill, and due to the aloofness and mystery which has always surrounded it, a computer is a computer.

With the advent of the microprocessor, however, this public perception of computers began to change. The availability of the personal computer as a consumer item initiated tremendous strides towards public enlightenment of the computer as a useful, entertaining tool. Now, computer stores are everywhere, hobbyists clubs exist in almost every major city, and news coverage and public visibility is swelling on a national, even worldwide, basis.

But it was not always like this. When Dick and Lois Heiser pioneered the first computer store in southern California in the fall of 1975, every move was a gamble. How was the public going to react? Was the initial surge due to a sincere interest, or due to a shortlived "fad." After just beginning to recover from a nationwide recession, would the public climb upon an expensive, yet unproven movement? And the equipment! Could it be obtained? Would it work? Could it be kept working? While the popular computer concept was dawning, the task remained to acquaint and convince the public of its existence and stability. Sure, there were a few computer clubs around, but they were technically slanted away from the typical consumer. Already existing, too, were the professional computer societies, but they had (and still have) no room for the novice. Some popular computer publications were around, but very few people knew of these. And buying a computer by mail-order? Unassembled? My face still pales at the thought!

Where oh where, then, was the solution? What structure could tie down the loose ends, and present a united, uncomplicated front to the consumer? Obviously, (NOW its obvious) the retail computer store. The revolution was underway!

A retail computer store? At first the idea seems incredible, but why not? The equipment is available, it is priced at levels that consumers can afford, and the applications are endless. With this challenge in mind, and with the four partners crossing their fingers, the enterprise known as The Computer Systemcenter came to life. Being an unprecedented concept, there were no guidelines to be used in formulating this marketing strategy, no viable way to survey public attitudes, no anything but talent, enthusiasm and ideas. Although possibly not the best and certainly subject to change, herein is contained a summary of our philosophy, methods, and initial experience in this endeavor.

WHY OPEN A RETAIL COMPUTER STORE

This, of course, was the first and foremost question to be resolved. Although no member of the partnership had ever claimed being pioneers to any movement, the following reasons were used to justify the adventure:

- To create public awareness that solid state technology has reduced the size, delicate nature and cost of computers while greatly increasing their reliability. Cost reductions, of course, demand the greatest emphasis.

**The Computer Systemcenter, 3330 Piedmont Road NE, Atlanta, Georgia 30305. (404) 231-1691.*

- To present this low-cost computer technology in a setting with which the public is comfortable and well acquainted. The open marketplace arrangement, with freedom of movement and selection, should be as advantageous with computer products as it is, for example, with stereo or photographic gear.
- To bring together under one roof the preferred items from the equipment lines of several manufacturers. In the old computer world to even suggest crossing the lines of one manufacturer with another is almost sacrilegious.
- To provide a local and publicly accessible source of computer expertise. The public, particularly in its initial learning stages, is going to be seeking a qualified technician with a layman's delivery.
- To generate public understanding, confidence, and most of all, enthusiasm about the coming revolution in popular computers. This obviously is a challenge and a responsibility. The psychological factors encountered by springing computers in everyone's face may well be endless, so we'll hold back any discussions on this.
- To be an innovator in a concept which will alter the lives and thinking of this generation and all following generations. Hopefully, too, this innovation will turn us a tidy reward in our generation. Innovations alone do not pay bills.

Okay, so we were convinced. And luckily so were enough investors to get the project off the ground. We then turned to the next question.

WHAT SHOULD A RETAIL COMPUTER STORE BE?

This should certainly be viewed as no small matter. After all, it's not everyday that one can set the philosophical guidelines for a revolutionary (there's that word again!) new concept. In addition to the normal properties any complete retail establishment should exhibit, such as good management, products and reputation, the unique nature of the computer as a product demands additional efforts and responsibilities. These include:

- A much higher level of competence. The retail computer dealer must become a cross between a personable department store clerk and a seasoned computer scientist (with a touch of showmanship a definite plus).
- An ability to communicate and to educate. Rarely will the typical consumer know less about what the product is and does than he will in this environment.
- An understanding of public apprehension and misconceptions about computers. Unfortunately, although through no real fault of their own, computers have received a lot of bad publicity and blame. And too, the "Big Brother" or "Numbered Society" image probably lurks in the back of almost everyone's mind.

MOVING FROM IDEAS TO REALITY

The notions of the armchair philosopher are good only to a certain point. It soon became time to move from concepts into a working concern. Actually establishing a real live computer store involved dozens of mundane questions that had to be answered and literally hundreds of small problems to solve, not to mention a few humdingers. Here is the rationale behind some of our major decisions.

- WHERE SHOULD THE STORE BE? We felt that the crucial aspect of our business was presenting computer technology to the people. Therefore, we had to be situated in a location with high visibility and accessibility. We finally chose a location in a new, small shopping center in northeast Atlanta.

- WHAT SHOULD THE PHYSICAL LAYOUT BE? Our store is broken into four sections. The public access is, of course, into the showroom where we continually maintain at least one complete system up and running for inspection and demonstrations. Equipment assembly, checkout, storage and repair is performed in what we call our work room, which is essentially a technicians shop. Then, for solitude and to retain some semblance of mental stability, we have our personal offices. Although tiny in size, this has not been a particularly important factor since we have had little time thus far to sit down. The last, and possibly most unique section of our establishment, is a large combination classroom/conference room. As a conference room, this area has had more or less predictable uses. However, due to the nature of our product, we felt that a classroom was mandatory for public instruction, seminars, and lectures.
- WHAT'S IN A COMPUTER SHOWROOM? In deciding what products to carry at the store our principle decision was which major line of computers to support. We felt that we could only offer adequate support and services for a single type of mainframe. Also, our supplier would need to offer the computers in both kit and assembled form for a reasonable cost. Lastly, the production and delivery capabilities of the manufacturer would need to adequately support us as dealers. At the time we made our decision, only one company seemed to fulfill these requirements. That was MITS of Albuquerque, New Mexico with the Altair 8800 line of computers. This decision was of course only a beginning, because even though MITS had several peripherals available for the Altair, we could not offer a complete product line strictly using their equipment. Thus we began making arrangements for various terminals (hardcopy, CRT, and color graphics) as well as several "plug compatible" devices such as interfaces to a standard television. We also felt it necessary to carry as many of the publications of interest to the general computer user as possible. Examples of these include *BYTE, Peoples Computer Company,* and *Creative Computing.* In the software area, we offer system programs such as monitors, assemblers, and text editors as well as a BASIC language. Computer games is an area of great interest to the consumer, and lends itself to entertaining but informative demonstrations. Consequently much of our initial effort has been in that area.
- HOW ABOUT SERVICES AND SUPPORT? Having the necessary equipment in the necessary place only serves as part of the attraction for any retail operation.

The ability to develop, troubleshoot, and maintain hardware and software is a must, as is the ability to speak (or at least listen) intelligently about pertinent matters on a consultation basis. And again, due to the very nature of our product, user community support is of utmost importance. The public must understand or feel that they will be given the chance to understand. We offer, for example, free introductory classes (both hardware and software) to the purchasers of our units, with only a nominal charge to non-purchasers. For the do-it-yourselfer, an hour (at least) each evening is set aside for him to bring in his under-the-weather Altair and get free troubleshooting. Other community functions include the active support of the local microcomputer hobbyist club, one of the largest of its kind in the nation. We also have gained the reputation as being a local depository of technical computer-related brochures and publications. Keeping abreast of the latest price changes and new product offerings is necessary for our survival.

INITIAL EXPERIENCE

October and November of 1975 were spent building and furnishing the store. During this period, we noticed a faint quickening of the public pulse at the shopping center where we are located. More and more people stopped by to talk and find out what type of place this was going to be. In November we were already working out of a half-finished store front. Finally, after months of preparation, we opened the doors on December 20, 1975. Since then the experiences have been truly remarkable. We have run the gamut from uproarious laughter to the utter frustration that seems destined to accompany any business operation. Problems? They occur by the dozens. Basically, however, they can be classified into one of two areas; either problems that are common to all small businesses, or problems unique to computer stores.

The largest obstacle we have had to overcome is our own lack of business experience. Initially this didn't seem too important, but since then we evolved our own form of Murphy's Law: If something can be done wrong—we will do it wrong; and just to be sure, we'll do it wrong two different ways. We have certainly not been immune from the various small ailments that plague small businesses — lack of management expertise, supply problems, cash flow, bad checks, you name it. At times these daily problems seem to outweigh and overwhelm everything else, causing us to occasionally have to reach down deep and rely on a certain humor to see us through. One of our pet diversions is coining "Anti-Slogans" that seem to fit the mood. We have a few classics, such as:

"Progress — We Sneer At The Term"
"Problems Are Our Most Important Product"
"Where Concepts Become Confusion-And Confusion Becomes A Way Of Life."

The other issues with which we deal are those unique to computer stores. First, there is the basic task of letting people know what we're trying to do. To the average person who walks in off the street, we usually have to tell them that even though they can be used as such, we're not selling calculators. Then we have to expect two stock questions, "What kind of place is this?", and "Well, what can you do with these computers?"

At first we would stammer around trying to pull together good answers, but by now it's practically a conditioned response. We hear one of these questions and bang! Put the old mind into AUTO and crank up the song and dance routine. I mean, we've got it down pat!

To characterize our typical customer is impossible. Applications range from monitoring water levels in the depths of a sewer, to writing payroll checks, to controlling a model railroad in someone's basement. Users include extremely sophisticated systems programmers as well as complete computer novices. Actually, it's less frustrating dealing with a complete novice who is somewhat awed by computers than it is to deal with an IBM 370 programmer who views microcomputers as "Toys." When this happens (and it does happen), we just take them in our computer room and show the business system on which we perform our accounting and inventory control (Altair 8800A, 40K of memory, dual disk units, video terminal and printer, all built into a custom desk). It's almost frightening when you think it's all based on a $30.00 microprocessor.

Our biggest miscalculation seems to have been just how much time is required by the computer novice. We tend to forget just how much there is to know about computers until we try to explain things to someone who thinks that a terminal is actually the computer. We've literally spent hours passionately pleading the case of Microcomputers to someone only to hear "Well, I'm really only in here killing time while my wife is shopping."

And the joys of Kit-building. Ah, there's a story in itself. Someone buys a Kit, puts it together overnight, it doesn't work, he screams, and brings it in muttering "damn crappy equipment." Usually, the next thing we hear is "What do you mean, bad solder joints? I went to the NASA soldering school." Still, we have a certain obligation to help each customer get his system up and running. We've tried to accomplish this by setting aside a certain time each day, (6:00-7:00PM), during which we have a free software and hardware clinic. During this time anyone can bring in their sick machines and/or programs and we'll give them a hand.

The latest issue we've had to deal with is the "software vacuum." People are discovering that after the machine is working, the real uses are just beginning. Canned programs are fine (programs written and debugged by someone else), but when it comes to writing one's own programs — well, there's more to software than meets the eye. To combat this situation, we have started a series of programming lectures entitled, The Art Of Creative Computer Programming." This series is aimed at providing a novice programmer with insights about programming and a set of software tools and tricks to tackle his own programming project.

WHERE DO WE GO FROM HERE

In a year's time computer stores have evolved from a few timid, rather speculative ventures to a firmly established concept. The first generation of stores are highly individualized with each having a different emphasis. In filling out the scorecard on ourselves, I would have to say that we set some very idealistic, but unrealistic, goals. But, there's no substitute for experience, and even with somewhat altered goals, our enthusiasm and energy still runs high.

What about the overall industry? The approaches offered by the differing stores are quite varied. At one extreme is the store that attempts to act primarily as a computer supermarket, emphasizing a broad assortment of equipment for the customer. The other extreme is taken by stores that emphasize primarily their service and support. Of course, this is actually a continuum. As computers become easier to use, as the general public becomes more aware, and as the software vacuum is filled, the tendency will be to move towards the supermarket concept. In these early stages, however, education, service, and installing consumer confidence must be paramount to all other considerations.

Would we do it over again? You can bet your solid-state bippies that we would. Each of our successes, whether a simple home computer or an intricate industrial system, causes feelings of pleasure and accomplishment. The dream of readily available computer power is now becoming a reality and we are sharing, and hopefully helping, in the transition.

RETAIL COMPUTER STORES

ARKANSAS

COMPUTER PRODUCTS UNLIMITED
4216 West 12th St.
Little Rock, AR 72204
(501) 666-2839

ARIZONA

TRI-TEK, INC.
6522 North 43rd Ave.
Glendale, AZ 85301
(602) 931-6949, 931-4528

CALIFORNIA

APPLIED COMPUTER TECHNOLOGY
1038 Merced
Berkeley, CA 94707
(415) 527-6760

BARGAIN ELECTRONICS
2018 Lomita Blvd.
Lomita, CA 90717
(213) 539-2260

BITS N' BYTES
1216 West Wilshire Ave.
Fullerton, CA 92633
(714) 525-9613

BYTE SHOP NO. 1
1063 El Camino Real
Mountain View, CA 94043
(415) 969-5464

BYTE SHOP NO. 2
3400 El Camino Real
Santa Clara, CA 95050
(408) 249-4221

BYTE SHOP NO. 3
2559 So. Bascom Ave.
Campbell, CA 95008
(408) 377-4685

BYTE SHOP NO. 4
1225 Ocean St.
Santa Cruz, CA 95060

COMPUTER CENTER
8205 Ronson Rd.
San Diego, CA 92111
(714) 292-5302

COMPUTER COMPONENTS, INC.
5848 Sepulveda Blvd.
Van Nuys, CA 91411
(213) 782-7924

COMPUTER KITS
1044 University Ave.
Berkeley, CA 94710
(415) 845-5300

THE COMPUTER MART
2333 Beverly Blvd.
Los Angeles, CA 90057
(213) 484-2002

COMPUTER MART OF LA
625 West Katella No. 10
Orange, CA 92667
(714) 633-1222

COMPUTER MEDIA, INC.
10090 N. Blaney, Suite 6
Cupertino, CA 95014
(408) 867-0885

THE COMPUTER STORE
820 Broadway
Santa Monica, CA 90401
(213) 451-0713

COMPUTER STORE OF SAN FRANCISCO
1093 Mission St.
San Francisco, CA 94103
(415) 431-0640

COMPUTER WAY, INC.
15525 Computer Ln.
Huntington Beach, CA 92649
(714) 892-8816

COMPUTERS AND STUFF
664 Via Alamo
San Lorenzo, CA 94580
(415) 278-4720

CTI DATA SYSTEMS
3450 East Spring St.
Long Beach, CA 90806
(213) 426-7375

THE DATA CENTER
c/o Programma Consultants
3400 Wilshire Blvd.
Los Angeles, CA 90010

DIALECT
1076 El Dorado Dr.
Livermore, CA 94550
(415) 433-0390

IV DIMENSION, INC.
7060 Miramar Rd. Suite 104
San Diego, CA 92121
(714) 566-7610

METATIC CORP.
2211 Fountain Oaks Drive
Morgan Hill, CA 95037
(408) 779-8150

MICRO BYTE
c/o Guy Hall
183 East 8th Ave.
Chico, CA 95926

MICROPROCESSOR MARKETING
c/o Tom Hudson
28120 Peacock Ridge Dr. No. 806
Rancho Palos Verdes, CA 90274

PETE'S ELECTRONICS
3007 Ventura Blvd.
Oxnard, CA 93030
(805) 485-6467

PROKO ELECTRONICS
975 Foothill
San Luis Obispo, CA 93402
(805) 544-5441

RAINBOW ENTERPRISE
10723 White Oak Ave.
Granada Hills, CA 91344
(213) 360-2171

SANDRLY ASSOC.
c/o Dickinson
7020 Balboa Blvd.
Van Nuys, CA 91406

THE SMALL BUSINESS COMPUTER CO.
400 Dewey Blvd.
San Francisco, CA 94116
(415) 665-2575

SUNNY SOUNDS
927 B E. Las Tunas Dr.
San Gabriel, CA 91776

SUNSHINE COMPUTER CO.
9 Palomino Lane
Carson, CA 90745
(213) 830-8965

COLORADO

COMPUTER COUNTRY, INC.
18 Alameda Square
2200 West Alameda
Denver, CO 80223
(303) 935-1100

GATEWAY ELECTRONICS
2839 W. 44th Ave.
Denver, CO 80211
(303) 458-5444

INTERMOUNTAIN DIGITAL
c/o Douglas M. Woodard
1027 Dellwood Ave.
Boulder, CO 80302

J.B. SAUNDERS CO.
3050 Valmont Rd.
Boulder, CO 80301
(303) 442-1212

CONNECTICUT

THE COMPUTER STORE
63 So. Main St.
Windsor Locks, CT 06096
(203) 627-0188

HEURISTIC SYSTEMS
c/o Susan Gilpatrick
244 Crystal Lake Rd.
Ellington, CT 06029

FLORIDA

THE COMPUTER STORE
c/o Comprehensive Systems
P.O. Box 251
Pensacola, FL 32502
(904) 434-6754

DOUGLAS COMPUTER SYSTEMS
710 Oaks Plantation Drive
Jacksonville, FL 32211
(904) 724-8726

ELECON CORPORATION
4981 - 72nd Ave. North
Pinellas Park, FL 33465
(813) 541-3021

MARSH DATA SYSTEMS
5405-B Southern Comfort Blvd.
Tampa, FL 33614
(813) 886-9890

MICROCOMPUTER SYSTEMS, INC.
144 So. Dale Mabry Hwy.
Tampa, FL 33609
(813) 879-4301

SUNNY COMPUTER STORES, INC.
117 Newton Road
West Hollywood, FL 33023
(305) 989-1258

GEORGIA

ATLANTA COMPUTER MART
5091-B Buford Highway
Atlanta, GA 30340
(404) 455-0647

THE COMPUTER SYSTEM CENTER
3330 Piedmont Rd.
Atlanta, GA 30305
(404) 231-1691

ILLINOIS

CHICAGO COMPUTER STORE
517 Talcott Rd. Hwy 62
Park Ridge, IL 60068

ITTY BITTY MACHINE CO.
1316 Chicago Ave.
Evanston, IL 60201
(312) EAT-6800

INDIANA

DATA DOMAIN
111 So. College Ave.
Bloomington, IN 47401

KENTUCKY

CYBERTRONICS, INC.
312 Productions Court
Louisville, KY 40299
(502) 459-0426

LOUISIANA

EXECUTONE MICROCOMPUTER
6969 Titian Ave.
Baton Rouge, LA 70806
(504) 383-1371

MARYLAND

THE COMPUTER WORKSHOP
5709 Frederick Ave.
Rockville, MD 20852
(301) 468-0455

MASSACHUSETTS

COMPUTER MART
473 Winter St.
Waltham, MA 02154
(617) 890-0677

THE COMPUTER STORE, INC.
120 Cambridge Street
Burlington, MA 01803
(617) 272-8770

THE COMPUTER STORE, INC.
P.O. Box 2621
Framingham Center, MA 01701
(617) 877-6984

MICHIGAN

COMPUMART, INC.
254 South Wagner Rd.
Ann Arbor, MI 48103

THE COMPUTER STORE OF ANN ARBOR
310 E. Washington St.
Ann Arbor, MI 48108

MISSOURI

GATEWAY ELECTRONICS
8123-25 Page Blvd.
St. Louis, MO 63130
(314) 427-6116

NEBRASKA

WELLING ELECTRONICS
529 North 33rd St.
Omaha, NE 68131
(402) 342-6564

NEW JERSEY

COMPUTER MART OF NJ
151 Kline Blvd.
Colonia, NJ 07067
(201) 574-2173

THE COMPUTER ROOM
451 Simons Ave.
Hackensack, NJ 07601
(201) 342-6667

HOBOKEN COMPUTER WORKS
56 Second St.
Hoboken, NJ 07030
(201) 949-2493/420-1644

TECHNICAL DESIGN LABS, INC.
342 Columbus Ave.
Trenton, NJ 08629
(609) 599-1868

WILLIAM ELECTRONIC SUPPLY
1863 Woodbridge Ave.
Edison, NJ 08817

NEW MEXICO

THE COMPUTER SHACK
3120 San Mateo NE
Albuquerque, NM 87110

NEW YORK

AUDIO DESIGN ELECTRONICS
487 Broadway, Rm 512
New York, NY 10013
(212) 226-2038

COLLEGIATE AUDIO
2A Telfort St.
Oneonta, NY 13820
(607) 432-1930

COMPUTER CORNER
White Plains Mall
200 Hamilton Ave.
White Plains, NY 10601
(914) 457-7853

COMPUTER ENTERPRISES
P.O. Box 71
Fayetteville, NY 13066
(315) 637-6208

COMPUTER MART OF NEW YORK
314 5th Ave.
New York, NY 10001
(212) 279-1048

THE COMPUTER STORE OF NEW YORK
55 West 39th St.
New York, NY 10018
(212) 221-1404

ELECTRONIC PRODUCTS
645 Stewart Ave.
Garden City, NY 11530

SYNCHRO SOUND ENTERPRISES
193-25 Jamaica Ave.
Hollis, NY 11423
(212) 359-1489/468-7067

OHIO

ELS SMALL COMPUTER CENTER
2209 Taylor Road
Cleveland Heights, OH 44112
(216) 249-7820

RIDGWAY EAST
Retail Computer Center
161 Bell St.
Chagrin Falls, OH 44022
Cleveland - (216) 247-4845

OREGON

THE REAL OREGON COMPUTER CO.
205 West 10th
Eugene, OR 97401

PENNSYLVANIA

THE COMPUTER ROOM
Juniper and Sansom Sts.
Philadelphia, PA

J.B. INDUSTRIES
610 West Olney Ave.
Philadelphia, PA 19120

LECTRO MEDIA, LTD.
22 New Market
2nd & Pine Sts.
Philadelphia, PA 19147
(215) 925-9090

MARTIN J. O'BOYLE ASSOC.
P.O. Box 9094
Pittsburgh, PA 15224
(412) 361-1602

PERSONAL COMPUTER CORP.
Frazer Mall
Frazer, PA 19355
(215) 647-8460

RHODE ISLAND

COMPUTER POWER, INC.
60 Harding St.
West Warwick, RI 02893
(401) 822-0204

TENNESSEE

BYTE'TRONICS
5604 Kingston Pike
Knoxville, TN 37919
(615) 588-8971

TEXAS

DIGITEX
2111 Farrington
Dallas, TX 75207
(214) 744-1849

ELECTRONIC SPECIALTY CO.
2006 San Sebastian B-232
Houston, TX 77058
(713) 475-4402

THE MICRO STORE
c/o David Wilson
2602 D. Custer Parkway
Richardson, TX 75080
(214) 231-4088

POLARIS COMPUTER SYSTEMS
3311 Richmond, Suite 200
Houston, TX 77006
(713) 527-0348

UTAH

COMPUTERS & STUFF
1092 S. State St.
Orem, UT 84057
(801) 224-2066

VIRGINIA

COMPUTER HOBBIES UNLIMITED
9601 Kendrick Rd.
Richmond, VA 23235
(804) 276-5056

MEDIA REACTIONS, INC.
11303 South Shore Drive
Reston, VA 22090
(703) 471-9330

MICROSYSTEMS
6605 A Backlick Rd.
Springfield, VA 22150
(703) 569-1110

WASHINGTON

MICROCOMPUTER APPLICATIONS
6009-B 13th Way SE
Lacey, WA 98503
(206) 456-3924

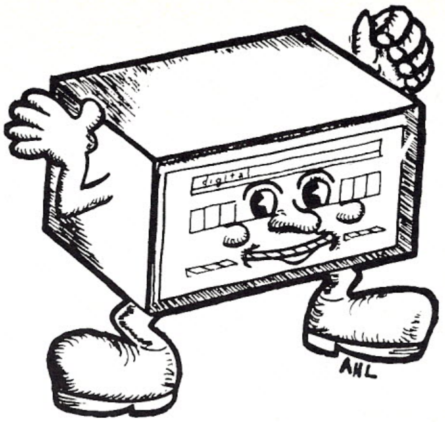
RETAIL COMPUTER STORE
410 N.E. 72nd
Seattle, WA 98115
(206) 524-4101

WISCONSIN

LAUERMAN DEPARTMENT STORE
Marinette, WI 54143
(715) 735-3311

MILWAUKEE COMPUTER STORE
3415 N. 49th St.
Milwaukee, WI 53216
(414) 445-2403

If you know of a computer store which is not listed here, or if the listing of a store is incorrect, please send us the correct listing. We will run updates every issue and the complete directory twice a year.

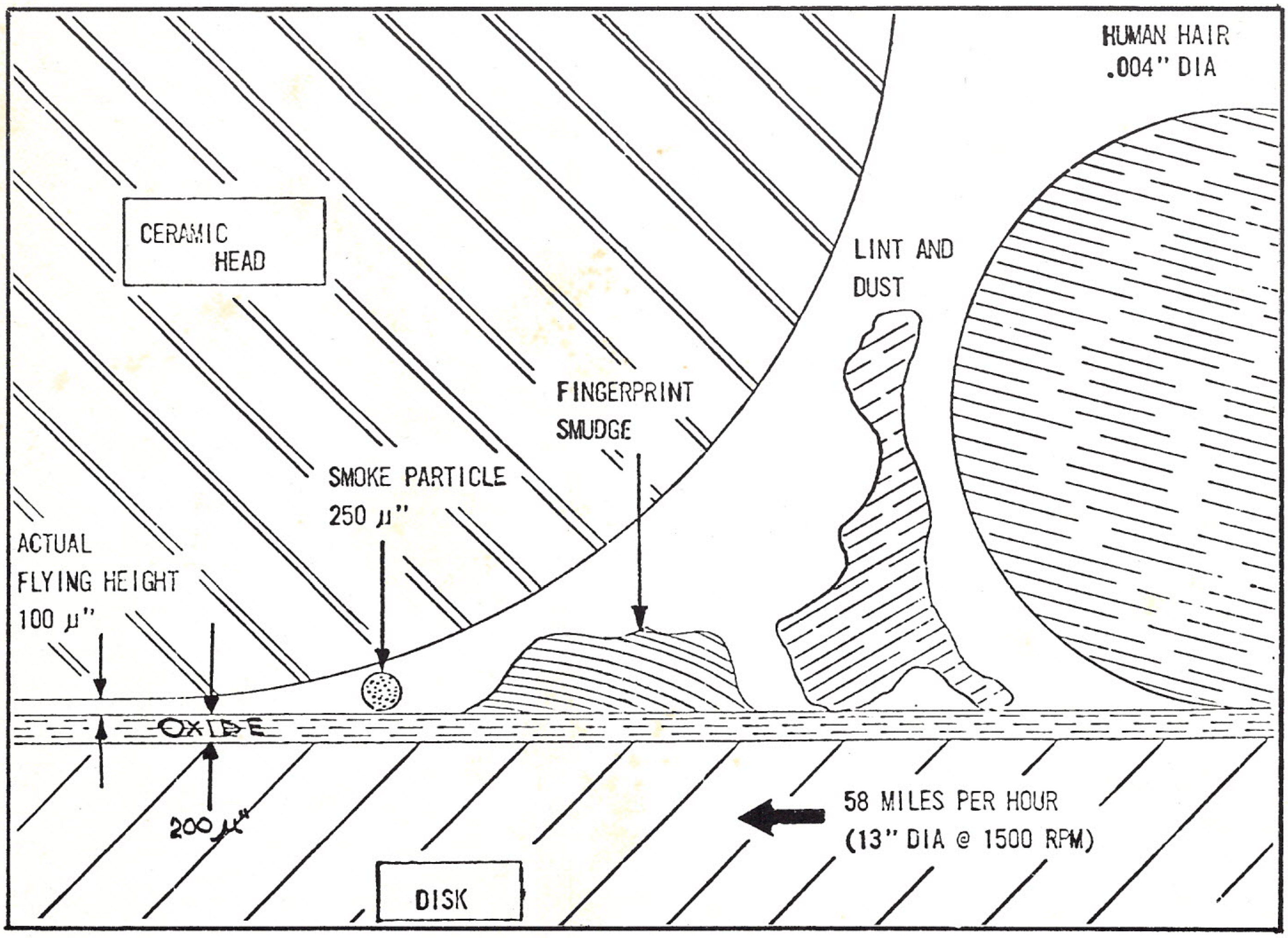


"COMPUTER CRITTER"
(Be sure to collect the complete set of 12 !!)

DISK DESTRUCTION MADE SIMPLE

by Bill Thorne, London, England

The diagram below vividly illustrates the effects that lint, dust, finger prints, smoke particles, etc., can have when they are on a disk surface and meet flying heads of a typical cartridge disk drive. The results, as you might imagine, can be very damaging to reliable disk performance. By storing disk cartridges more carefully and keeping them in a clean environment, reliability can be dramatically improved.



What really happens when a major computer in a modern information network breaks down?

Charles Mosmann

PULLING THE PLUG

"All this business about computers taking over is nonsense, of course, but it's dangerous nonsense. It's just not true. Newspapers trying to scare up a headline."

The noise level in the coffee shop had suddenly abated, as though everyone paused to take a breath at once. I couldn't help overhearing the remarks being made at the table behind me.

"Right," a second voice affirmed, "You can always pull the plug."

I did not turn around to see who the two men were; they were undoubtedly a couple of the hundreds, perhaps even thousands, of engineers and programmers who seemed to be involved in virtually every enterprise in the large office building where I worked.

When I got back to my office, I kept thinking about the idea of pulling the plug. Did anybody really know where the plug was? In fact, modern information networks are designed so that if one computer stops working, the work load is automatically shifted around to some other machine. The computer that accepts a user's job is likely to be no more than a scheduling and switching center, which will locate a computer somewhere else to do the actual processing.

I had a few minutes before I was going to have to leave for a meeting. I decided to pursue the question a little further to pass the time. I have a typewriter-like terminal beside my desk which is my own personal access to the computer network. I turned my chair to face it and flicked on the switch. The computer system to which I subscribe announced itself and asked me for identification. I typed in my name, my account number, and a password known only to the system and me, which functions as a sort of combination lock so that no one else can mess with my files or run up my bill, pretending to be me.

Finally, all these formalities were taken care of satisfactorily and the machine typed out, "READY." The following dialogue took place between us.

```
READY

    list available socio-economic models.

ECO-75, CAL81, USECON, ECOMETRIC, USECON2.
MORE?

    no. call CAL81.

(pause)

YOU HAVE CALLED CAL81, A SIMULATION OF LOCAL
AND NATIONAL SOCIO-ECONOMIC SYSTEMS. DO
YOU NEED A DESCRIPTION OF CURRENT FILES AND
CAPABILITIES?

    no.

PLEASE STATE PRESUPPOSITIONS OF YOUR SIMULA-
TION.

    time - current. status - normal except: pull the plug
    on this computer.
```

DEFINE "THIS COMPUTER." DEFINE "PULL THE PLUG."

this computer - the computer on which CAL81 is now running and which is in communication with user 15a-225/85. pull the plug - this computer stops operating and does not restart.

OK. PLEASE STATE OPERATION/OUTPUT REQUIREMENTS.

significant changes to systems/this city/next 7 days.

(pause)

DAY 1, 14:07. MASTER COMPUTER 307/21 AT COMPUTIL INC LOSANGELES CEASES OPERATION. NO REASON REPORTED. NO ESTIMATE OF DOWNTIME AVAILABLE. ALL UNCOMPLETED JOBS ROUTED TO COMPUTIL SANDIEGO UNTIL FURTHER NOTICE.

DAY 1, 16:30. SUDDEN INCREASE IN WORKLOAD AT BANK CLOSING TIME OVERLOADS SANDIEGO COMPUTER AND EXCESS ROUTED VIA LOSANGELES TO SANFRANCISCO.

DAY 1, 16:37. UNANTICIPATED POWER SURGE IN SANFRANCISCO TEMPORARILY INCAPACITATES COMMUNICATIONS SATELLITE COMPUTER. WORK IS SENT BACK TO LOSANGELES WHICH ATTEMPTS TO REROUTE IT TO DENVER.

DAY 1, 16:38. SYSTEM DEADLOCK IN DENVER COMPUTER DUE TO SUDDEN OVERLOAD ON SCHEDULING ROUTINE. NO OLD JOBS ARE COMPLETED. ALL NEW JOBS ENTERING SYSTEM ARE ROUTED TO LOSANGELES OR SANDIEGO.

DAY 1, 16:39. LOSANGELES COMMUNICATIONS CENTER OVERLOADED AND SOME MESSAGES ARE LOST.

DAY 1, 17:00. CITY BANKS CANNOT BALANCE OUT ACCOUNTS. AT FOUR BANKS THERE ARE NO PROVISIONS FOR CLOSING THE BANK UNDER THESE CONDITIONS. PERSONNEL STANDING BY AWAITING INSTRUCTIONS.

DAY 1, 17:00. THREE LARGE MANUFACTURING PLANTS IN LOSANGELES AND TWO IN SANDIEGO USING COMPUTER-CONTROLLED TIME CLOCKS. THESE ARE RUNNING SLOW AND WILL NOT ALLOW WORKERS TO GO HOME.

DAY 1, 21:15. MANUAL PROCEDURES FINALLY SUCCEED IN CLOSING DOWN BANKS AND PLANTS FOR THE NIGHT.

DAY 2, 09:00. 23 PAYROLLS DUE THIS MORNING ARE NOT AVAILABLE DUE TO CRISIS OF DAY BEFORE. BANKS RECEIVE NUMEROUS REQUESTS FOR SHORT-TERM CREDIT WHICH HAVE TO BE DENIED BECAUSE CREDIT FILE IS NOT AVAILABLE, DUE TO FAILURE TO CLOSE OUT BOOKS PROPERLY DURING CRISIS OF DAY BEFORE.

DAY 2, 11:40. OVERLOAD ON SANDIEGO CENTER CAUSES DECISION BY ELECTRIC UTILITY TO SHIFT OPERATION OF GENERATOR SCHEDULING PROGRAM TO PHOENIX COMPUTER. MOMENTARY DIP IN POWER LEVEL DURING THIS SHIFT.

DAY 2, 11:41. DIP IN POWER LEVEL DAMAGES DISC STORAGE AND DISC DRIVE OF SANDIEGO MASTER COMPUTER. TAPE RENDERED UNREADABLE ON INPUT/OUTPUT COMPUTER.

DAY 2, 11:42. AUTOMATIC SCHEDULING PROGRAM AT SANDIEGO COMPUTER CENTER DECIDES TO LOWER WORKLOAD BY TRANSFERRING IT TO SEATTLE VIA TELEPHONE. DELAY IN CARRYING OUT PLAN DUE TO FAILURE OF LONG-DISTANCE BILLING SYSTEM; ALL LONG-DISTANCE CALLS MUST BE PLACED MANUALLY.

DAY 2, 12:00. DAMAGED DISC IN SANDIEGO CONTAINED UNIQUE COPY OF YESTERDAY'S POLICE FILES. 17 PRISONERS RELEASED BECAUSE THERE ARE NO RECORDS, NO GROUNDS FOR DETAINING THEM.

DAY 2, 14:00. DAMAGED TAPE HAD CREDIT RECORDS OF UNITED TRUST AND LOAN COMPANY. ALL LOAN OPERATIONS TEMPORARILY SUSPENDED; ALL ESCROW OPERATIONS DELAYED AT LEAST ONE WEEK.

DAY 3, 09:00. 26 RETAIL STORES CLOSE TODAY FOR INDEFINITE PERIOD, CONSEQUENCE OF SEVERE CASH/CREDIT SHORTAGE.

DAY 4, 09:00. ALL FUEL DELIVERIES BY NATIONAL STANDARD OIL COMPANY TO THEIR DISTRIBUTORS ARE HALTED. THEIR SIMULATION AND TRANSPORTATION SCHEDULING SYSTEM WAS BUMPED FROM COMPUTER BY PRIORITY JOB FROM CITY MANAGER'S OFFICE.

DAY 4, 15:00. EVENING EDITION OF DAILY TIMES DEMANDS FULL INVESTIGATION OF CRISIS! TO BEGIN IMMEDIATELY. ALSO ANNOUNCES THAT PUBLICATION WILL BE HALTED IMMEDIATELY AS OF THIS DATE, SINCE AUTOMATIC TYPESETTING SYSTEM CANNOT BE RUN. BLAME PLACED ON SEVERE SHORTAGE OF COMPUTING CAPABILITY IN THE CITY.

DAY 5, 09:10. UNITED TRUCKING, OPERATIONS TRANSPORTATION, AND AGRI-TRUCK INC FORCED TO CURTAIL OPERATIONS DUE TO FUEL SHORTAGE AND CASH/CREDIT CRISIS.

DAY 5, 10:15. REPRESENTATIVES OF LOCAL MARKET CHAINS HOLD PRESS CONFERENCE, INDICATING MAJOR FOOD SHORTAGES. SOME EVIDENCE OF HOARDING ALSO REPORTED.

DAY 5, 12:10. CITIZENS MASS AT CITY HALL, DEMANDING ACTION. COMPLAINTS INCLUDE: LOSS OF JOBS, SCARCITY OF FOOD, ALARMING DIP IN QUALITY OF TV PROGRAMMING DUE TO COMPUTER SHORTAGE.

DAY 5, 16:00. UNDERGROUND NEWSPAPER NOW ONLY PAPER IN CITY STILL IN PUBLICATION, DUE TO PRIMITIVE MANUAL NATURE OF ITS OPERATION. THIS WEEK'S EDITION PLACES TOTAL BLAME FOR CRISIS ON "GIANT BRAINS" AND CALLS FOR DESTRUCTION OF COMPUTERS.

DAY 6, 07:00. POLICE REPORT THAT NUMEROUS COMPUTERS AND OTHER AUTOMATIC MACHINES HAVE BEEN DAMAGED DURING THE NIGHT. PUBLIC ELECTRIC COMPANY HAS TO SHUT DOWN THREE GENERATORS; CALLS ON NEIGHBORING DISTRICTS TO HELP OVERCOME EMERGENCY.

DAY 6, 07:30. DRAIN ON NEIGHBORING DISTRICTS CAUSES INTERMITTENT POWER FAILURES THROUGHOUT THE STATE.

DAY 6, 09:00. POWER FAILURE CAUSES ALL COMPUTERS IN STATE CAPITOL TO STOP OPERATING, EXCEPT FOR MILITARY COMPUTERS WHICH RUN ON EMERGENCY GENERATORS.

DAY 7, 09:30. GOVERNOR REPORTS ALL SYSTEMS IN STATE ARE NOW INOPERATIVE EXCEPT FOR NATIONAL DEFENSE.

END OF 7-DAY REPORT. CONTINUE?

no. status report, national level, as of day 10.

PRESIDENT DEMANDS TOTAL ISOLATION OF STATE TO AVOID DOMINO EFFECT AND SYSTEMATIC DESTRUCTION OF NATIONAL LIFE-STYLE.

successful?

NO. TELEPHONE, TELEGRAPH, RAIL, COMMERCIAL TRUCK, AIR TRANSPORTATION ARE CLOSED DOWN. MICRO-WAVE INTER-COMPUTER NET HAS FAILSAFE PROVISION WHICH CAUSES IT TO REFUSE TO ACCEPT TRAFFIC BECAUSE OF DANGER OF OVERLOAD. COMPUTERS IN NEIGHBORING STATES BEGIN TO SHUNT POWER/INFORMATION SERVICES TO DAMAGED AREAS.

effect?

NEIGHBORING STATES EXPERIENCE SIMILAR BREAKDOWNS. PRESIDENT DECLARES NATIONAL EMERGENCY AFTER BRIEF CABINET MEETING.

effect of this declaration?

MINIMAL. NO MEANS TO PROMULGATE MESSAGE.

status of population?

DEATHS FROM STARVATION AND CIVIL INSURRECTION. DETAILED STATISTICS?

no.

CONTINUE SIMULATION OR TERMINATE?

terminate.

SIMULATION TERMINATED. SAVE FILE OR SCRAP FILE?

scrap.

FILE SAVED. SIGN OFF.

My curiosity had been satisfied by this brief game. I checked my watch, saw that it was time to leave for my meeting, and turned away from my terminal. As I was putting papers in my briefcase, I heard the rapid clicking noise of a further message being typed. I did not bother to check what it was; not infrequently, systems information of only marginal importance is printed out for users who may be interested.

As I was putting on my coat and turning to the door, a funny thing happened. The ceiling lights in the room were beginning to flicker. I instinctively turned my head up to look at them. A few of the flourescent panels were already dark. I don't know what made me look down at the terminal just then, but I did. And in the fading light I saw the message that had been printed a few minutes before:

MASTER COMPUTER 307/21 AT COMPUTIL INC LOS-ANGELES CEASES OPERATION. NO REASON REPORTED. NO ESTIMATE OF DOWNTIME AVAILABLE. ALL UNCOMPLETED JOBS ROUTED TO COMPUTIL SANDIEGO UNTIL FURTHER NOTICE. END OF MESSAGE.

Before I took my eyes from the page, my office was totally dark.

POEMS

by Peter Payack

THE COSMIC CLOCK

Life crawled
out of the sea,
three hundred million
years ago.

On the cosmic clock
that's only a few
hours.

The cosmic clock
runs on pure energy,
with stars for
jewels.
It's a one hundred
thousand million
billion jewel clock,
which hasn't been
rewound since
the big bang,
15 billion years ago.

It wasn't
made in Switzerland,
either.

The old man
down the street
wears it on
a gold chain.

It shines
in the dark
like cat's
eyes.

INTUITION

A flip-flop is female,
Built of two NAND gates
Whose output is fed back
To each other as input.

When the input to both
NAND gates is positive,
The flip-flop is in
The indeterminate state.

This does not mean
It cannot decide whether
To go the one way or the other,
But only that you
Cannot predict ahead of time
Which way it will go.

Electrons are tricky
In flip-flops,
Like they were in the lead
Which kept the Alchemists
Guessing.

Flip-flops are built of
NAND gates, and NAND gates
Are powered by electrons,
Which can fool you, just like
They fooled Paracelsus.

PERIOD

According
to Einstein's
theory
of relativity,
any given
point
in space
and time
can be declared
the center
of things.

The period
at the end
of this poem
is the center
of the universe,
now.

THE WHITE LINE

A white line
divides the road

like the border
between sanity and madness.

I drive on the right side
and chuckle at those who come
in the opposite direction:
"Insane bastards!"

How much longer will a computer illiterate be considered educated? How long will he be employable and for what jobs? Is it enough to be merely a subject of computer administered instruction?

Should the computer teach the student, or vice-versa?

by ARTHUR W. LUEHRMANN
Dartmouth College
Hanover, New Hampshire

This sermon begins with a parable.

Once upon a time in the ancient past there was a nation in which writing and reading had not yet been invented. Society was as advanced as possible, considering that it had no mechanism for recording the letter of the law or of writing agreements, contracts, or debts. Nor was there a way of recording the heritage of information and knowledge that had to be passed on from generation to generation.

As a result, a great fraction of the total effort of the society was spent in oral transmission of information. Master teachers, who themselves had been taught by older master teachers, lectured before children and young people of the society. Training a master teacher was a long and expensive process, and so the society could not afford many. For reasons of economy the curriculum was quite rigid and lectures were on a fixed schedule. Teaching, obviously, was a labor-intensive industry based on skilled, expensive talent. Education, per force, was a luxury that could be afforded by the elite classes only.

Then, one day, writing and reading were invented. Not surprisingly, the first application of this new technology was to business and government. Money was printed; laws were encoded; treaties were signed. In response to these needs, a reading and writing industry grew up. Within a few years it was able to offer a broad range of reading and writing services to its customers. The customers found this to be a convenient arrangement, since hiring readers and writers from service vendors eliminated the need for each customer to invest in an expensive R&D effort of its own. The customers remained illiterate.

At first the situation was somewhat chaotic. Each vendor of reading and writing service tended to develop its own favorite language and its own technique for encoding information, leading to incompatibilities that impeded the spread of the new technology. After a winnowing-out period, however, the number of competing systems settled down to a few and major difficulties were handled by translators—though inevitably something seemed to be lost in the process.

Always looking for new markets, the vendors of reading and writing service began to examine the area of education. In view of its elitist role in the society it had been dismissed at first as too limited a market. A few, more imaginative people, however, argued that the application of reading and writing technology could turn education into a mass market. They proposed the following plan of attack. Reading and writing specialists and master teachers would work as a team. The master teachers would deliver their best, most carefully prepared lectures to the reading and writing experts, who would write them carefully *verbatim* into books. The books would then be copied many times, and each copy would be made available to a new type of educational functionary—the *reader*. His only job would be to assemble groups of students and to read aloud to them the recorded lectures of the master teachers. In view of the fact that training such a reader would be far less expensive than the education of a master teacher, the on-going cost of such a program would be far less than that of the conventional lecture method. The new method came to be called Writing Assisted Instruction, frequently abbreviated to WAI.

Needless to say, WAI had its opponents. Established master teachers expressed doubt whether a less skilled reader would be able to communicate subtleties of inflection, and they were certain that a mere reader could not process student responses with skill or intelligence. WAI proponents counter-charged that the master teachers were merely expressing their vested interest in the present educational establishment, and, indeed, that they ought to be fearful because the superiority of WAI would ultimately drive out the conventional practitioners. Even within the education establishment some younger members became WAI supporters on the grounds that the new method was a boon to education research. Until then, teaching had

Reprinted from Proceedings of Spring Joint Computer Conference, 1972. Copyright 1972 by AFIPS.

been something of a black art, shrouded in the privacy of the classroom. To compare one teacher with another was impossible. But in the future, they said, the written record of the lectures of master teachers would make the teaching experience explicit and subject to analysis, comparison and improvement. It was high time, the young Turks exclaimed, that the teaching profession act with accountability to the public it served.

Unfortunately, such controversy remained for many years on a hypothetical plane. The number of actual WAI efforts was very small and their results were not striking. There was also a credibility problem. Many of the most outspoken advocates of WAI, especially in the legislature and in business and on local school boards, were themselves almost totally illiterate in the new reading and writing skills. How could they evaluate a new technology if they had not mastered it themselves?

Finally, government, business and some members of the education establishment decided to mount two or three large-scale demonstrations of WAI in order to show publicly the advantages of the new educational technology. For a period of several years curriculum experts collected information on a few key courses of lectures by assorted master teachers. The reading and writing experts wrote down the best series and read them aloud to the curriculum experts, who would criticize them and make improvements. The reading and writing experts would then incorporate the improvements in the next draft. Then came the field test. Readers began to read the drafts aloud to actual classes of students, and this led to further revision by the curriculum experts and rewriting by the reading and writing experts. At the end of a few more years a summative evaluation of the projects was undertaken by an independent, reputable educational testing organization, whose mission was to compare the cost and effectiveness of WAI with conventional education.

The parable is nearing its conclusion now. Actually it has two alternate endings, one happy and one sad. The sad ending, which follows now, is brief.

The educational testing organization reported that the projects were a complete vindication of Writing Assisted Instruction. It found that students taught by WAI performed even better on standardized tests than students taught by the average master teacher, that the students liked WAI better, and that the total cost of WAI was about a fourth that of conventional instruction. These pilot projects were imitated on a grand scale and education was revolutionized. Special institutes turned out vast numbers of readers and within ten years they were reading courses of lectures aloud to masses of people who could never have been educated before the new instructional technology arrived. The nation grew and prospered and thanked the day that the reading and writing industry was founded.

That is the sad ending. The happy ending is somewhat longer and more complicated. Here it is:

The educational testing organization found that WAI was neither measurably worse than conventional instruction, nor better. It found that costs were somewhat higher than anticipated, mainly because the market demand for people with reading and writing skills had driven their wages up near those of master teachers.

But this lukewarm finding was anticlimactic when it came, for the impact of reading and writing on education had taken a new turn during the intervening years. Here is how it happened.

At first a few master teachers had themselves found it necessary in pursuing their own research to spend the enormous effort required to master the skills of reading and writing. As they became more and more competent readers and writers, they began to see clearly the power of the written word within their own disciplines. Naturally enough the humanists were the first to apply this new intellectual tool to their fields of interest. Literature specialists collected stories, wrote them down, exchanged them with each other and began to develop literary criticism to a new height. Language specialists compiled lists of grammatical rules, which became writing manuals. Scientists were slower in becoming literate, with mathematicians leading the way, since they grasped the possibility of writing mathematical concepts in abstract notation. Nevertheless, for many years scientists continued to remain in verbal darkness.

While reading and writing had its primary impact on scholarly research, at the same time many master teachers across the land began to wonder whether it might not be beneficial to introduce elementary uses of reading and writing to students in their courses. A few language teachers began to show students how to write phrases and sentences, and the more venturesome teachers even asked students to write sentences of their own. Such experience, they claimed, greatly enhanced a student's understanding of syntax and rules of grammar. Even in subject areas far removed from language, to which reading and writing have a natural affinity, teachers began to report pedagogical gains due to having students carry out elementary reading and writing tasks as an adjunct to conventional instruction.

One obstacle to student use of reading and writing was the awkwardness of the main systems of notation, which had been developed mainly for research and business applications. The most popular such system was particularly difficult to format, since its characters all had to be positioned accurately in a fixed number of columns. Occasionally there were rumors that a group of teachers in a remote province near the northern frontier had developed a simpler writing system and all their students were using it daily. Such rumors were hard to verify; only a few people ever voyaged that far north, and, in any case, experts in the reading and writing industry seemed confident that anything that made the current system simpler would also take away its power and elegance. So most teachers adhered to it.

Within a few years teachers began to hold national meetings to tell one another how their students used reading and writing within their courses. Advocates of this type of use, which came to be called *adjunctive*, insisted that it be distinguished clearly from WAI. Writing Assisted Instruction, they charged, was nothing more than an improvement in the technology of delivering instruction. Adjunctive use of reading and writing by the student, on the other hand, represented a change in the intellectual content of instruction. They argued from the following philosophical premise:

> Reading and writing constitute a new and fundamental intellectual resource. To use that resource as a mere delivery system for instruction, but not to give a student instruction in how he might use the resource himself, was the chief failure of the WAI effort, they said. What a loss of opportunity, they exclaimed, if the skill of reading and writing were to be harnessed for the purpose of turning out masses of students who were unable to read and write!

WAI advocates responded that it was well and good that a few elitist schools teach their students the difficult skill of reading and writing; it was enough that WAI teach lesser skills to masses that might otherwise remain uneducated and unemployable.

How much longer, asked the WAI opponents in rebuttal, will an illiterate person be considered educated? How long will he be employable and for what jobs if elitist schools are turning out competent readers and writers by the hundreds?

The more visionary advocates of mass literacy told of foreseeing the day when students would spend more hours of the day reading and writing than listening to lectures. Small research libraries had indeed sprung up at some schools, but they were expensive operations limited to a few specialists who had to raise funds to pay for their use. Such people were particularly incredulous at the suggestion that every school ought to adopt as an educational goal the establishment of a significant library open freely to all students. School administrators were at first appalled at the idea that the library should not be on a pay-as-you-go basis but should be budgeted as part of the general institutional overhead costs.

But as time went on and even school administrators became competent and imaginative users of the skill of reading and writing, all schools gradually accepted as a mission the bringing of literacy to all students. Accreditation agencies examined the quality of libraries before approving schools. Books began to appear all over and finally even in people's homes. WAI did not die out altogether, but continued as a cost-effective alternative to the lecture. But as books reduced dependence on lectures, students made less use of both WAI and lectures and spent more time on their own reading and writing projects. The nation grew and prospered and wrote poems in praise of the day that reading and writing were discovered and made available to all people.

End of parable.

It is a perilous strategy, bordering on bad taste, to tell a joke and then for several pages explain why it was supposed to be funny. However, this allegorical tale has been told here not merely for entertainment but mainly for the moral lesson it carries. To compare reading and writing with computing might be dismissed as an amusing frivolity; but that would be wrong. Our fundamental philosophical premise here is that, like reading and writing,

> "[computing] constitutes a new and fundamental intellectual resource. To use that resource as a mere delivery system for instruction, but not to give a student instruction in how he might use the resource himself, has been the chief failure of the [C]AI effort. What a loss of opportunity if the skill of [computing] were to be harnessed for the purpose of turning out masses of students who were unable to [use computing]!"

As this example shows, it is a trivial editing task to go through the entire reading and writing fable and turn it into a story about computing and its uses in education. In fairness, the author admits that the story really *is* about computing and that reverse editing was done in the original telling so that it would seem to be about reading and writing. Yet, as a story about reading and writing it has considerable plausibility, doesn't it? The Writing Assisted Instruction program outlined in the story is not a totally absurd idea for putting reading and writing to use in education. One cannot argue against claims that committing lectures to writing would make education available to more people, would invite critical comparisons and a consequent improvement in subsequent revisions of written materials, and would be an asset to the study of the learning process itself. What does appear absurd, however, is the failure of these mythical WAI proponents to recognize that the best educational use of reading and writing is the teaching of reading and writing itself to everyone. Mass literacy is an educational mission about which few of us have doubts today.

Yet that consensus among us seems to vanish when one substitutes "computing" for "reading and writing" and "CAI" for "WAI". Mass computing literacy is not an agreed-upon educational goal. Today very few courses at any educational level show students how to use computing as an intellectual tool with applications to the subject matter being taught. Oh, there are a few isolated, subject-matter-free courses in computer programming; but their market is largely restricted to vocational-education students, at one end of the spectrum, and future computer professionals at the other. It is true that most schools consider it prestigious to have a large and powerful computer facility; but the fact of the matter is that such computers are usually the captives of research and administrative interests and operate on a pay-as-you-go basis. Ironically, it is in the most prestigious universities that students are least likely to be permitted to use those prestigious computers. It is a rare secondary school, college, or university that budgets and operates its computer facility in the same way that it budgets and operates its library. (There is a persistent rumor of an exceptional example in some remote province near the northern frontier, but so few people ever travel that way that the report is hard to verify.) In the main, literacy in computing simply is not an educational goal at many schools. Most educators seem to find bizarre the suggestion that accreditation agencies examine schools for the quality of their educational computing facilities, just as they now do with libraries.

The distressing truth today is that educators, local school boards and federal policy-makers are far more receptive to the plans of CAI proponents for using the technology of computing as a cost-effective delivery system for instruction in math or remedial English than they are to making computing itself a part of education. This statement should not be taken as a blast against CAI. On the contrary, CAI advocates are to be commended for their desire to reduce the cost of instruction, to tailor it to the different learning styles of students, to develop systems that encourage closer examination of what is being taught and systems for improving instruction, and to hold teachers and schools accountable to their clientele. With enough developmental work on CAI, it is likely that students will perceive the computer as a very superior teacher. Above all, CAI promises to make education a less labor-intensive industry and so to enable masses of people to become better educated. This is certainly a goal worth working for.

But there is a higher goal. If the computer is so powerful a resource that it can be programmed to simulate the instructional process, shouldn't we be teaching our students mastery of this powerful intellectual tool? Is it enough that a student be the subject of computer administered instruction—the end-user of a new technology? Or should his education also include learning to use the computer (1) to get information in the social sciences from a large data-base inquiry system, or (2) to simulate an ecological system, or (3) to solve problems by using algorithms, or (4) to acquire laboratory data and analyze it, or (5) to represent textual information for editing and analysis, or (6) to represent musical information for analysis, or (7) to create and process graphical information? These uses of computers in education cause students to become masters of computing, not merely its subjects.

It will be countered that such an educational mission is well and good for a few elitist schools, where students are willing to learn the difficult skill of computing; but it is enough that CAI teach lesser skills to masses of students that might otherwise remain uneducated and unemployable.

In response we ask, how much longer will a computer illiterate be considered educated? How long will he be employable and for what jobs if elitist schools are turning out competent computer users by the thousands?

The true story about computing and education is at its midpoint. Like the reading and writing parable, it has a sad ending and a happy ending. Which one actually occurs will be determined by you—teachers, school administrators, computer professionals, and government policy-makers.

©CREATIVE COMPUTING

"Now hear this! *I* am the programmer. You are the *programee*!

Opinion

THE GOVERNMENT DINOSAUR

Charles Winn
530 McAlway Road
Charlotte, N. C. 28211

Over the years we have witnessed the growth and development of what could best be described as the governmental dinosaur — a large unwieldy critter, who though well-meaning, thinks slowly, moves awkwardly, reacts tardily, and is beset with many ailments and infirmities. Like its animal form it may well ultimately become so ineffective and weighed down as to collapse into death and extinction. This governmental monstrosity has assumed duties and responsibilities out of all proportion to its intrinsic talents and abilities. It was designed or put together piecemeal to serve in slower less demanding times and has had new and challenging obligations thrust upon it.

Human comprehension and reaction at the governmental and legislative levels is incapable of coping with the multitude or complexity of rapidly changing facts, figures, and demands. New discoveries, advanced technologies, and other newly revealed knowledge moves at such a rapid pace that it defies human assimilation and coordination by those who require these benefits and insights most. Government, like the great retarded beast, stumbles and staggers its way along — trampling many underfoot, wavering from a proper course, over or under compensating, and becoming more lost and confused. The mere size of the organism renders it ineffectual. The multitudinous bureaus, departments, offices, agencies, and divisions grow like cancerous cells in wild profusion. When they do function or respond it is not always in conjunction with other parts or in their best interests. Communication and cooperation between these segments is a nightmare activity.

Other investigators and writers have examined and ably described the wild inconsistencies and glaring deficiencies of this tremendous organization. We are all familiar with some of the errors, shortcomings, and the ineffectiveness of the government as it exists today. It is no one individual's fault that these conditions prevail, but does this ignorant beast in its present form warrant our trust and confidence? Does it even begin to efficiently serve the potential and actual needs of the American populace?

This beast is not peculiar to only the United States, but has its counterparts in other countries. In fact, there the colossus may be even more ponderous, awkward, and intractable. Our chief concern however, is with remedies, improvements, and innovations. This process shouldn't require revolution in anything but our thinking and our solutions.

Initially, we might try a massive undertaking — the creation of the largest, most efficient computer yet conceived or assembled — a computer capable of retaining, assimilating, and processing information relating to all phases of governmental concern or interest. The machine would require a sizeable complex of experts to insure its proper function. Initially, to insure the accuracy of data, statistics, and information gathered, there would have to be a virtually fool-proof systems organization to gather such data and a board of experts from various fields and disciplines, men above suspicion or reproach and of sterling character, to make final judgments on what material was to be programmed and to oversee those that physically entered information into the machine. The margin for error would most certainly be less than it is under present random circumstances.

The machine could be programmed to temporarily reject material that was inconsistent with or deviant from information already retained. This material, in turn, could be reviewed by the board to ascertain its accuracy before resubmission. Computer experts and designers could build the necessary safety procedures and safeguards into the machine at its inception to prevent tampering and misuse.

This computer might well be dubbed Uncle Sam and could be relied on for accuracy, rationale, and impartiality. The machine may not be infallible, but would be infinitely superior to the mass confusion and human error that exist today. It could provide the president and his cabinet, the Congress, and other essential government figures with up-to-date and comprehensive information on which decisions could be partially based. The extent to which this information would be acted upon could be determined by experience and performance. Uncle Sam would obviate much of the reliance placed upon outside lobbyists, who are almost always selfishly motivated, and self-appointed experts whose information is often, at best, questionable. Uncle Sam could offset and reduce much of the mediocrity, partiality, and outright chicanery that now exists in the government area.

If these machines can be relied upon to deliver expensive and complicated space vehicles to obscure destinations with unerring accuracy and exactitude, they could certainly be channeled to the task of eliminating much of the human weakness and shortcomings, the boondoggery, and all of the prolonged hassling that they generate in government affairs. The legislative and administrative systems have become a snarl of ineptitude. The idea of Uncle Sam is not inconsistent with the human element for humans are, in fact, computers of a sort that lack the total recall, instantaneous operation, tirelessness, and the uncanny accuracy of their machine counterparts. They work very well in harmony and in conjunction with each other — the one providing what the other lacks.

The slowness and frequently the absence of reaction of the government to social needs results in the creation of a sense of alienation and helplessness in the general populace. Many experts and observers feel that the citizen's inability to express himself and make himself heard is the greatest weakness and ill in our society. Millions regard themselves as ineffectual pawns incapable of bringing about change or improvement even at the lowest levels of government and administration — they crave recognition and a sense of participation.

This pressing need might well be also answered by computer technology and provide a step toward truly democratic procedures. Present knowledge is sufficient to devise an electronic voting system whereby the average voter could be consulted and heard on basic issues. A coded card, similar to a credit card, might be issued to a qualified voter. This, in turn, could identify the voter upon insertion into an apparatus incorporated into the telephone system, thereby permitting the voter to dial his vote by following designated procedures. This type of equipment could be in every home possessing telephone service and conveniently located booths could serve others who lack individual telephone service or who may be away from their homes.

Radio and television could be more fully employed to keep the public abreast of current issues and prepared to make intelligent decisions on matters relating to the public welfare. Computers could tally these votes and furnish them directly to legislative government thus bringing about a return to democratic procedures unknown since the days of the great Grecian cities. If anything this would be improvement upon any democratic procedure ever employed on a large scale basis. For the first time the public could express itself directly without all of the representative inconsistencies and blindness that we now experience.

Another improvement would be to gradually transfer much of the governmental function to segments of private industry. This would be in keeping with our professed faith in free enterprise and consistent with the fact that business is much more efficient and goal oriented than the present framework of governmental administration.

These duties and responsibilities could still be originated, supervised, advised, and checked upon by government personnel at the upper levels. The functions and activities themselves would be the assignment of private companies or independent agencies that are geared for efficiency and results. This arrangement would produce greater accomplishment of goals and at great savings to the taxpayer. There would be benefits to be reaped all the way around. One such benefit would be the elimination of duplication and the removal of personnel who are often incapable, indifferent, or lacking in motivation or dedication. The present efficient government employee would have nothing to fear for he would readily be absorbed into the new system with extended responsibilities and a much greater opportunity to prove himself.

There is nothing sanctified or holy about governmental form that precludes its being changed or altered to conform to new demands and needs, to make better responses. Many of the founders of our country expressed this attitude freely and made allowances for it in the Constitution. If we're going to retain the old dinosaur, let's at least give it some assistance and provide it with modern aids. It deserves our wholehearted support and is most certainly in our best interests.

Juvenile Information System Killed

Last April, an unusual alliance of computer professionals and civil liberties advocates managed to halt the implementation of a Juvenile Information System in the Santa Clara, California Probation Department. The victory was the result of a lot of people learning about other people's problems and beginning to care about them.

Dorothy Ellenburg, director of the Council for Community Action Planning, Inc. in San Jose brought to light the problem with JIS. The system, which was to centralize the files of juveniles in order to enable agencies all over the county to have access to them, raised questions among her group because it called for the inclusion of the names of juveniles who weren't officially "delinquent". With the help of Carol Guddal, a technical writer for Hewlett-Packard and more than 30 other volunteer computer professionals, the CCAP studied the abuses to which the JIS might be subject, and were able to convince the county board of supervisors to defeat its implementation. A central objection to JIS that the computer professionals brought to light was its lack of security. With terminals all over the county that would produce not just displays of data, but printouts too, there was little that could be done to control the confidentiality of information. CCAP argued that such easy access to information about children who weren't even criminals would be tantamount to labelling them as "asocial kids". The computer experts who studied the implications of the system agreed.

A byproduct of the joint effort was the establishment of a data confidentiality commission in the county. The group is studying the implications of computerization and information sharing of the more than three million individual records compiled by county government on its citizens. Carol Guddal stated very clearly what she and the other computer experts learned and intend to remember as a result of their involvement: "It's easy to become isolated from the use of these systems after they've been shipped out the door. . . . I'm not so sure that I'll be guilty of it anymore. I never realized how profoundly a computer system can influence a community. They are not toys."

SYSTEMATIC TRI-PHASE PROJECTION

(Prophetic Rhetoric)

by Linda S. Labelson

Our company has integrated a transitional logistical concept. The total policy options consist of flexibility and capability with a functional parallelled time-phase projection.

Total hardware mobility is to be monitored with programmed digital synchronized contingency.

Our objective is third-generation programming; responsive, balanced management, with a totalitarian compatibility.

Circumspectly, we purport an accelerated expediting of this highly technical innovation.

"Computers and Beauty"

by Mutsuko SASAKI and Tateaki SASAKI*
1-18-17, Syoan Suginami-ku, Tokyo 167, JAPAN

In this article we consider beauty and simple methods of its realization by a computer. Here, by the beauty we do not mean such beauty as of logic or of love but mean artistic beauty appealing to the human's aesthetic sense.

In order to realize the beauty systematically by a computer, we need the concept of beauty to be defined suitably for programming. In searching for such a definition, we first note a very simple and very typical example: the "golden section," i.e., the most *beautiful* sectioning of a finite line into two parts. It is impressive that the ratio giving the golden section is not far from 1:1. If the ratio was 1:10, say, then the corresponding section would not appeal to our aesthetic sense. One of other typical examples is music; a beautiful melody would be an ugly one if the arrangement of notes in the music were bad. We can suppose from these examples an *essential* element of beauty; we may call it harmony. A thing being lacking in harmony will make the human's feeling unstable and cause an antipathy in his mind. Saying this from the viewpoint of beauty, we can say that it is not beautiful. On the other hand, we also note that the ratio giving the golden section is not exactly 1:1. We may describe this as that variety is necessary for the object being *more* beautiful. Let us illustrate this by cooking: cooking must contain the nutriment, but even the nutrimental cooking is not a better one if it tastes bad! Thus variety is an *additional* element of beauty. In this sense, an object which is full of variety but lacking in harmony, say a picture drawn by utilizing only random numbers, will not be beautiful. Therefore, one can adopt *the representation of harmony and variety* as a simple definition of beauty. It should be noted that both harmony and variety are conceptually much less abstract than beauty itself and suitable for computer programming because they can be converted into numerical relations rather easily. In the following we shall consider actual methods of realization of beauty through a computer by taking up the drawing as a concrete example.

For the sake of explanatory convenience, we classify the pictorial beauty into i) mathematical beauty, ii) natural beauty, and iii) creative beauty. By these terms we mean, respectively, that i) beauty found in figures easily expressible in terms of mathematical functions, ii) beauty found in nature, and iii) beauty being dependent mainly on the human's creative powers. Many of the actual drawings may, of course, contain two or all of these types of beauty. We are not sure that above classification covers all types of the pictorial beauty, but the reader will see that it is very adequate for the computer drawing.

i) We often experience that simple mathematical functions enable us to make excellently beautiful drawings. Most functions commonly used in applied mathematics represent continuous changes of one dependent variable with respect to changes of other independent variables under some definite rules. These rules maintain the balance of the resulting drawing as a whole, and continuity of the functions prevents the local configurations of the drawing from becoming too various. Thus we may say that most mathematical functions are harmonic. (The functions called harmonic in mathematics are solutions of the Laplace's differential equation. We are using the term "harmonic" in wider meaning here.) Perhaps the mathematical functions are best suited for expressing harmony simply. What we should care for in this case is how to represent variety. Fortunately, variety is also contained in the functions to some extent, and mathematics provides us with random numbers. Hence, we may be able to represent variety sufficiently in terms of mathematics. Many works of the computer art up to the present are based on the mathematical beauty. This beauty is so popular to computer artists and so easy to realize by a computer that we do not discuss it anymore.

ii) Most people will agree with an opinion that nature is full of variety. But we can also find a great many kinds of harmony in nature; e.g., figures of flowers, butterflies, fishes, mountains, etc., etc. We think this is because that laws in physics, biology, etc. govern the nature and make it be harmonic: matters are built from atoms systematically, only evolved and selected beings are surviving and they are balanced as a whole due to the struggle for existence, and even the nonliving things are balanced. We can hence represent harmony as well as variety by imitating the natural objects. Further the natural objects can also cause through our memory many psychological effects which are no more contained in the category of beauty. The important points in representing beauty by natural objects are that the resulting drawing should not become too complicated and that each object drawn in the drawing should be easily identified. The reason is that natural objects themselves are sufficiently various. For example, suppose we are drawing many flowers on a paper. If we only outline the flowers the drawing would not be so beautiful. If the overlapped and invisible parts are also drawn, the resulting drawing will be too complicated and may even be ugly. In this case, therefore, more complicated programs or peripheral systems are necessary than in the previous case. Some examples are an image reader and a picture recording system, but here we explain our simple program by showing illustrations. Figure 1 shows an input drawing for our system. The drawing showing three flowers in Fig. 2 is rather ugly because it is too complicated; even to identify it as representing three flowers is not so easy. Invisible parts are eliminated in Fig. 3. In Fig. 4 insides of the flowers are "painted" with assigned patterns. The resulting drawing is much more beautiful than that shown in Fig. 2. In this way, we can see that we can well represent beauty by imitating natural objects and processing them through even a simple system. We should of course design many systems in order to raise the computer art to higher positions. But, since nature is full of variety, we can conceive many possibilities. In particular, since we know beautifulness of the mathematical beauty and easiness of its handling, we have a good possibility in creating a new type of beauty which has not been well considered so far, by synthesizing the mathematical and the natural beauties.

iii) It is very common that artists raise aesthetic effects of their pictures by emphasizing the principal parts, changing color tones, modifying and distorting the figures to be drawn, and so on. Further artists often make pictures by using mainly their inspiration and imagination. The beauty represented in these cases is created by the human's creative powers. Therefore, in order to treat it systematically and automatically we

*Present affiliation: The Institute of Physical and Chemical Research, Wako-shi, Saitama 351, JAPAN.

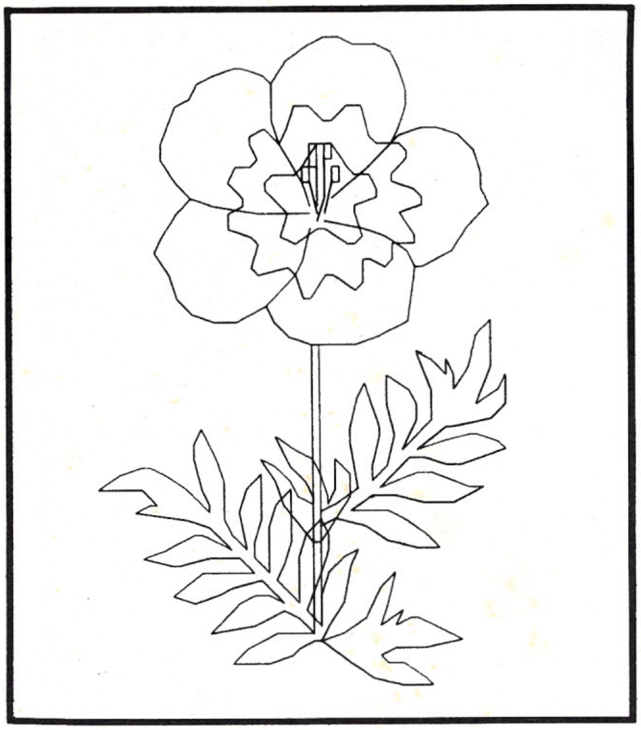


Fig. 1

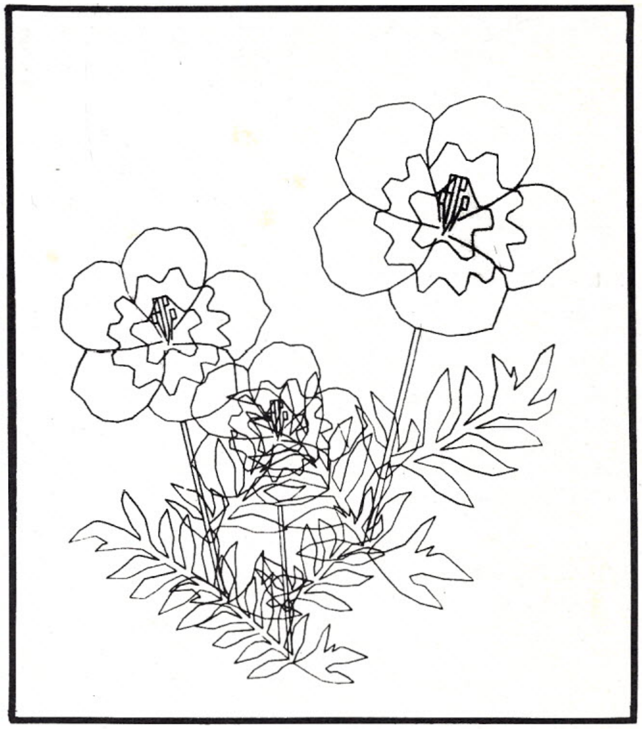


Fig. 2

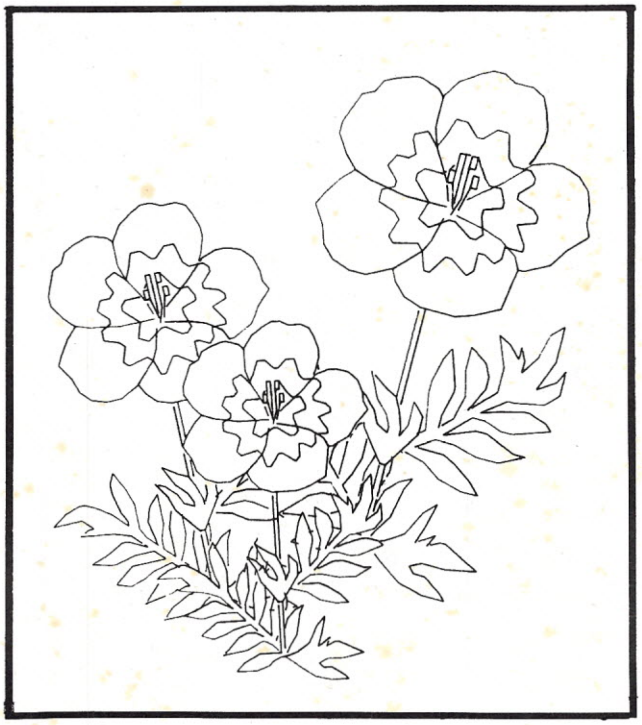


Fig. 3

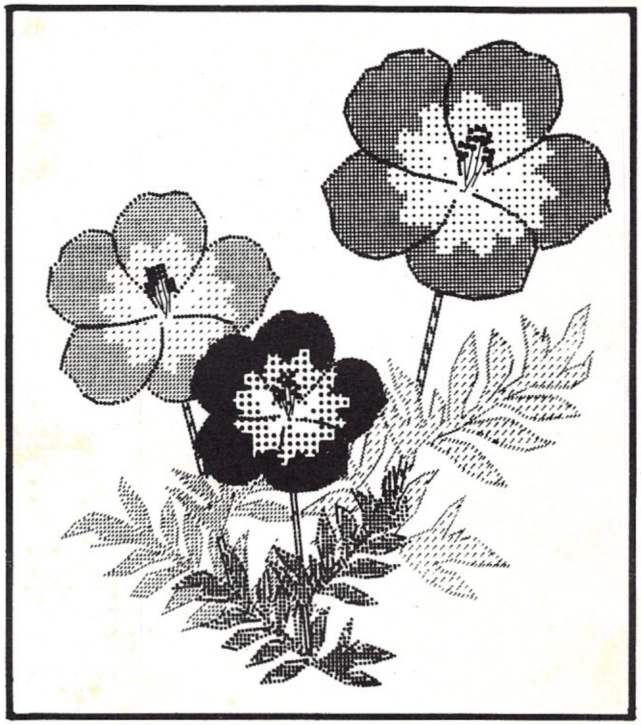


Fig. 4

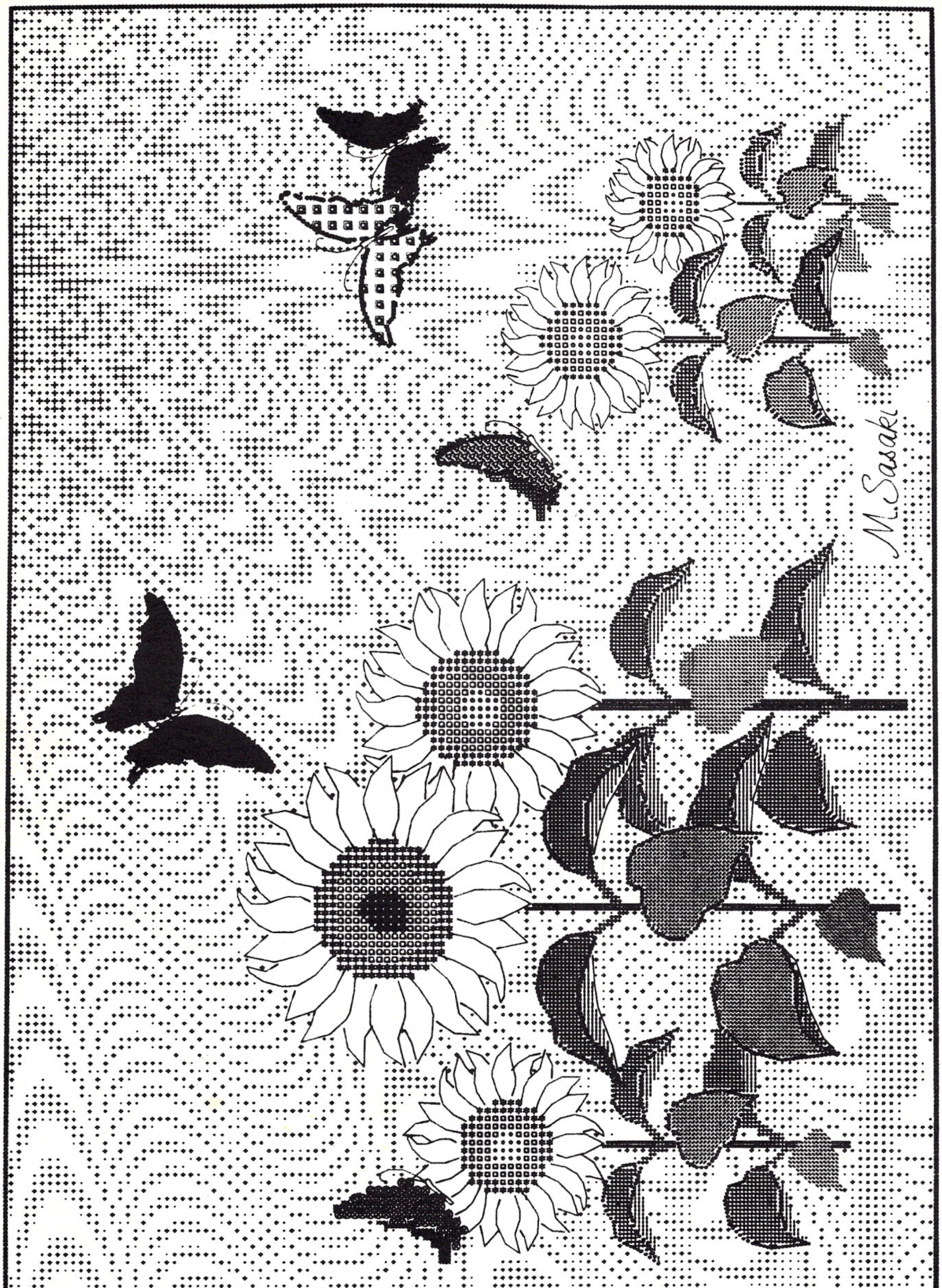
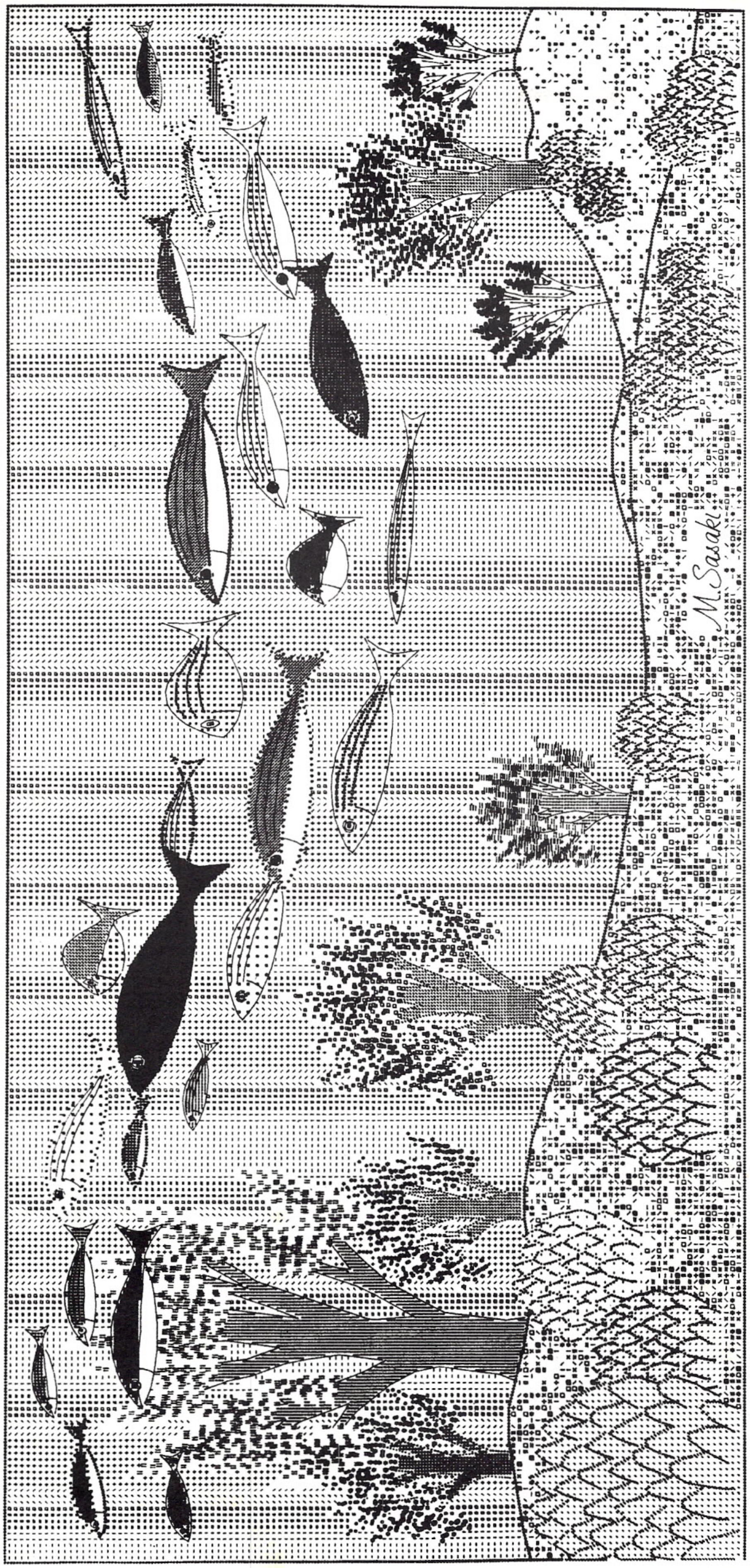


Fig. 5: "Hellow Sunflowers!, II." Original size 24 x 34 cm^2. Computer execution time 48 sec on a FACOM 230-75.



Perhaps we will have to educate and train the computer so that it can appreciate beauty.

must implement, more or less, the creative powers to computers. This is a very important task, because the creative powers are so much widely required in almost all of the art works; even in handling mathematical functions we encounter many problems whose solutions require the creative powers, such as how to use the functions in the drawing. On the other hand, the implementation is certainly a difficult task. We must investigate many things: What procedures are effective in creating beauty? How can we realize them by a computer? And so on. Perhaps we will have to educate and train the computer so that it can appreciate beauty, by making trials and errors. An important strategy which we would like to propose here is to ease the difficulty by adopting our definition of beauty and considering all things from this viewpoint. We have seen that the definition is persuasive and useful for the previous two types of beauty. We think it is also useful enough for the creative beauty. For example, the artist's techniques mentioned at the beginning of this section can be explained in part as that artists recover harmony from otherwise excess of variety. Of course there are many ways in representing harmony and variety, and much more detailed investigations are necessary in the actual programming. But above example seems to show usefulness of our strategy clearly. We can not say much about the realization of creative beauty through a computer at present, because we have not considered it yet. A few years later we will surely be able to say much about it.

Finally we show our recent works in Figs. 5, 6, and 7. They were produced by our system for computer art named ART-3, whose main functions are i) transformation of the input figures, ii) elimination of the invisible parts, and iii) to paint the surfaces by various patterns. ART-3 contains more than fifty mathematical functions for the figure transformation and the pattern generation. We can see from these figures how ART-3 works and how mathematical and natural beauties are united by it.

←

Fig. 7: "Surmarine walk, II." Original size 24 x 50 cm^2. Computer execution time 83 sec on a FACOM 230-75.

Figure 6 appears on the cover.

LET US FIRST MAKE IT

or

AND NOW I SAW, THOUGH TOO LATE

or

ROBINSON CRUSOE: A BOOK FOR ALL COMPUTING SEASONS

by Robert Taylor
Teachers College, Columbia University

Upon re-reading *Robinson Crusoe* recently, I was struck by the numerous parallels between the activities of Crusoe on his island and the work of today's analysts and programmers. Crusoe laid plans; established timetables; implemented solutions, modified his implementations, and occasionally even abandoned them; provided backup for various system components; and so forth. One might even argue that he trained a junior programmer or analyst named Friday.

THIS I DID WITHOUT FIRE, BY MERE MALLET AND CHISEL

The book is so rich in readable case studies appropriate for today's analyst or programmer to ponder that any computing professional would do well to read (or re-read) it at his earliest convenience. Why should what many regard as the first novel in the English language be so appropriate to today's analysts and programmers? Because Defoe was a good story-teller and because human problem solving hasn't changed as much as we sometimes like to imagine. Consider just two examples from Defoe's text: (1) Robinson Crusoe's unsuccessful attempt to escape from his island by building his own dugout canoe and (2) his successful attempt to guard against the catastrophic loss of his goat herd. The first illustrates how need causes the desperate to overlook fatal design flaws and attempt impossible implementations. The second illustrates the design and implementation of a backup system.

Attempting to implement a catastrophically flawed system

The first illustration concerns Crusoe's attempt to build a dugout canoe with which he can escape from the island. Note that the seductive attractiveness of a portion of the system (building the canoe) clouds his critical insight into the major flaw of the overall system (launching the canoe once it's built).

> At length, I began to think whether it was not possible to make myself a canoe, or periagua, such as the natives of these climates make, even without tools, or, as I might say, without hands, of the trunk of a great tree. This I not only thought possible, but easy, and pleased myself extremely with the idea of making it, and with my having much more convenience for it than any of the Negroes or Indians; but not at all considering the particular inconveniences which I lay under more than the Indians did, viz., the want of hands to move it into the water when it was made, a difficulty much harder for me to surmount than all the consequences of want of tools could be to them: for what could it avail me, if, after I had chosen my tree, and with much trouble cut it down, and might be able with my tools to hew and dub the outside into the proper shape of a boat, and burn or cut the inside to make it hollow, so as to make a boat of it – if, after all this, I must leave it just where I found it, and was not able to launch it into the water?
>
> One would imagine, if I had had the least reflection upon my mind of my circumstances while I was making this boat, I should have immediately thought how I was to get it into the sea: but my thoughts were so intent upon my voyage in it, that I never once considered how I should get it off the land; and it was really, in its own nature, more easy for me to guide it over forty-five miles of sea, than the forty-five fathoms of land, where it lay, to set it afloat in the water.
>
> I went to work upon this boat the most like a fool that ever man did, who had any of his senses awake. I pleased

myself with the design, without determining whether I was able to undertake it; not but that the difficulty of launching my boat came often into my head; but I put a stop on my own inquiries into it, by this foolish answer: Let us first make it; I warrant I will find some way or other to get it along when it is done.

This was a most preposterous method; but the eagerness of my fancy prevailed, and to work I went. I felled a cedar tree, and I question much whether Solomon ever had such a one for the building of the Temple at Jerusalem; it was five feet ten inches diameter at the lower part next the stump, and four feet eleven inches diameter at the end of twenty-two feet, where it lessened and then parted into branches. It was not without infinite labour that I felled this tree; I was twenty days hacking and hewing at the bottom, and fourteen more getting the branches and limbs, and the vast spreading head of it, cut off; after this, it cost me a month to shape it and dub it to a proportion, and to something like the bottom of a boat, that it might swim upright as it ought to do. It cost me near three months more to clear the inside, and work it out so as to make an exact boat of it: this I did, indeed, without fire, by mere mallet and chisel, and by the dint of hard labour, till I had brought it to be a very handsome periagua, and big enough to have carried six-and-twenty men, and consequently big enough to have carried me and all my cargo.

When I had gone through this work, I was extremely delighted with it. The boat was really much bigger than ever I saw a canoe or a periagua that was made of one tree, in my life. Many a weary stroke it had cost, you may be sure; and there remained nothing but to get it into the water; which, had I accomplished, I make no question but I should have begun the maddest voyage, and the most unlikely to be performed, that ever was undertaken.

> But all my devices to get it into the water failed me; though they cost me inexpressible labour too. It lay about one hundred yards from the water, and not more; but the first inconvenience was, it was up hill towards the creek. Well, to take away this discouragement, I resolved to dig into the surface of the earth and so make a declivity; this I began, and it cost me a prodigious deal of pains; but who grudges pains that have their deliverance in view? When this was worked through, and this difficulty managed, I was still much the same, for I could no more stir the canoe than I could the other boat. Then I measured the distance of ground, and resolved to cut a dock, or canal, to bring the water up to the canoe, seeing I could not bring the canoe down to the water. Well, I began this work; and when I began to enter upon it, and calculate how deep it was to be dug, how broad, how the stuff was to be thrown out, I found by the number of hands I had, having none but my own, that it must have been ten or twelve years before I could have gone through with it; for the shore lay so high, that at the upper end it must have been at least twenty feet deep; this attempt, though with great reluctancy, I was at length obliged to give over also.
>
> This grieved me heartily; and now I saw, though too late, the folly of beginning a work before we count the cost, and before we judge rightly of our own strength to go through with it.

This passage gives a very graphic picture of how the seduction takes place. His decision to "first make it" and then to worry about launching the boat later characterizes all too well the misplaced optimism we all seem to succumb to from time to time in designing and implementing systems.

Backing up major system components

The second illustration concerns Crusoe's design and implementation of a backup supply of goats. Earlier in the narrative, Crusoe went to considerable pain to capture and domesticate some wild goats and they have since become a principal component in his diet.

> While this was doing, I was not altogether careless of my other affairs: for I had a great concern upon me for my little herd of goats; they were not only a ready supply to me on every occasion, and began to be sufficient for me, without the expense of powder and shot, but also without the fatigue of hunting after the wild ones; and I was loath to lose the advantage of them, and to have them all to nurse up over again.
>
> For this purpose, after long consideration, I could think of but two ways to preserve them: one was, to find another convenient place to dig a cave under ground, and to drive them into it every night; and the other was, to enclose two or three little bits of land, remote from one another, and as much concealed as I could, where I might keep about half a dozen young goats in each place; so that if any disaster happened to the flock in general, I might be able to raise them again with little trouble and time; and this, though it would require a great deal of time and labour, I thought was the most rational design.
>
> Accordingly, I spent some time to find out the most retired parts of the island; and I pitched upon one, which was as private, indeed, as my heart could wish for: it was a little damp piece of ground, in the middle of the hollow and thick woods, where, as is observed, I almost lost myself once before, endeavouring to come back that way from the eastern part of the island. Here I found a clear piece of land, near three acres, so surrounded with woods, that it was almost an enclosure by nature; at least, it did not want near so much labour to make it so as the other pieces of ground I had worked so hard at.
>
> I immediately went to work with this piece of ground, and in less than a month's time I had so fenced it round, that my flock, or herd, call it which you please, who were not so wild now as at first they might be supposed to be, were well enough secured in it; so, without any further delay, I removed ten young she-goats and two he-goats to this piece; and when they were there, I continued to perfect the fence, till I had made it as secure as the other, which, however, I did at more leisure, and it took me up more time by a great deal.

That Robinson Crusoe should provide backup for such an essential component of his survival as his goat herd seems so obvious as to need no comment. However, how often do professional programmers and analysts fail utterly to provide backup for equally essential components of their own systems?

Further reflections

Cases like the two just presented abound in the book. Mistakes and successes are even replicated, just as they too often are in the contemporary systems world. The insight which these cases afford is particularly appealing because the adventure does not purport to deal with systems or computing at all! Moreover, though the cases largely lack the complex human interface problems common to systems involving many people (Robinson Crusoe lives and labors alone on his island for most of the narrative), this simplification allows Defoe to strip problems to their essentials. Crusoe's work is done primarily for himself. *He is his own end user.* As such, he must, in the most literal of all senses, "live with" the system he creates. Defoe thus presents us with the entire systems environment in a very compact form.

What else lies within the narrative of Crusoe's life on the island? Get a copy of *Robinson Crusoe,* read it, and find out. Be prepared to see your systems work in a refreshing new light. And should you feel too critical of Crusoe's solutions to his many problems, notice how long he survived. His solutions were viable enough to keep him going for 27 years, 2 months, and 19 days. Can any of us top that?

WHAT IS COMPUTER LITERACY?

by David Moursund
University of Oregon

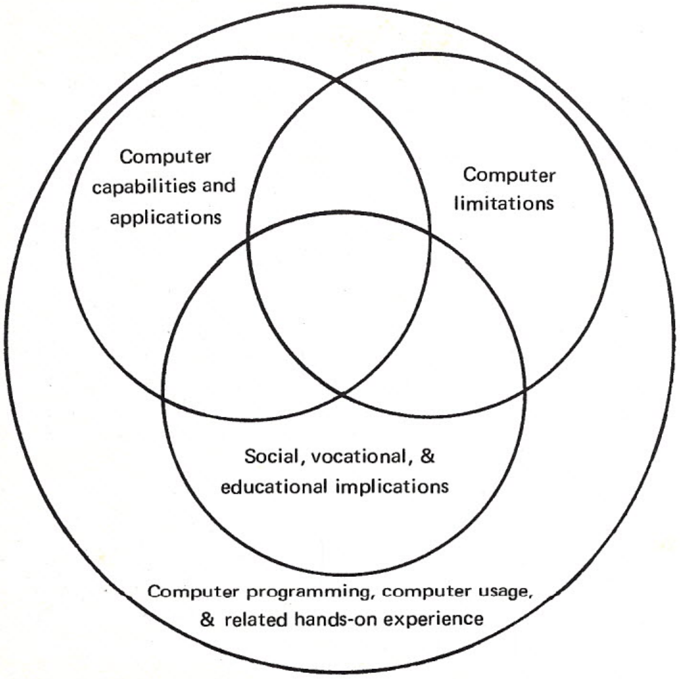
The concept of "computer literacy" is receiving much mention today. Over a period of time, we have developed a definition.

> Computer literacy refers to a knowledge of the non-technical and low-technical aspects of the capabilities and limitations of computers, and of the social, vocational, and educational implications of computers.

While such a definition can provide a focus for thought and discussion, it still does not pinpoint what is meant by computer literacy. Among other things it does not provide a measure of computer literacy nor a method for improving a person's level of computer literacy.

Most of you are familiar with the question "What is IQ?" and the answer "IQ is what is measured by an IQ test." It seems to me that we are at a similar stage of development for CL (computer literacy). Lately, many course outlines for computer concepts or computer literacy courses at the college level have been developed at Oregon and elsewhere. These courses are designed to raise a person's level of CL, and a knowledge of the content of such courses constitutes a certain level of computer literacy.

The University of Oregon's computer concepts course is a no-prerequisite, low level, introductory computer science course. Its major goal is to raise a student's level of computer literacy. Over a period of six years the course has evolved to the current point, where its content is approximately 1/3 computer programming and 2/3 non-programming materials. A Venn diagram of the course content is given below.


Computer capabilities and applications
Computer limitations
Social, vocational, & educational implications
Computer programming, computer usage, & related hands-on experience

In the diagram the computer programming, computer usage, and hands-on experience provides a foundation upon which the non-programming aspects of the course are built. Each of these four areas strongly overlaps the other three, and each supports the other three. A well balanced course needs aspects of each of these four areas.

It seems difficult to develop a course that is coherent and well integrated, and still preserves a reasonable balance among the four major areas. Probably the computer programming and related computer usage and hands-on experience is the major source of trouble. Most computer programming texts are designed to teach computer programming. That is, their major goal is to move a student rapidly along the computer programmer path. Most such books contain little information on the capabilities, limitations, or implications of computers. The material is not organized in a manner to make it fit in well with non-programming, computer literacy materials.

To overcome this difficulty in the UO's course, I have written a 150 page book, *BASIC Programming for Computer Literacy.* This book is currently being used in the course, and seems to be a satisfactory text. It is available for $4.00 (which includes postage and handling) from the Computer Science Department, University of Oregon, Eugene, Oregon 97403.

The non-programming content of a CL course can range over a wide variety of topics, and will depend to a certain extent upon the interests and knowledge of the instructor. One cannot tell if a person is computer literate on the basis of a single true-false or multiple choice question. That is, CL refers to a broad, integrated knowledge of low level computer science. Such knowledge must include many facts and how these facts interrelate. But it is difficult to isolate a single fact that is indispensable, or fundamental.

On the non-programming content of the course, I use an objective-type final exam. In fall 1974 this exam consisted of 150 questions. An item analysis was run on these questions to determine which were the more difficult and which best differentiated the students who scored high on the test from those who scored lower on the test. Thirty of the better questions (harder, and good differentiators) have been selected and appear at the end of this article. A student making an A or high B on the exam probably answered at least 3/4 of these questions correctly.

The answers in most cases are not obvious. The 30 question test was administered to students on the first day of the winter term 1975 course. The class average was 14.75. Random guessing by all students would have produced a class average of about 12.

Taken individually, the merits of any single question are certainly subject to debate. One can easily argue that the question is not relevant to his concept of what constitutes computer literacy. Taken as a whole, however, such a group of questions provides a reasonably broad measure of many parts of the non-programming content of a computer literacy course. Try the test yourself. Try it on your students. Individual questions can provide a good basis for class discussion or individual student reading/study projects.

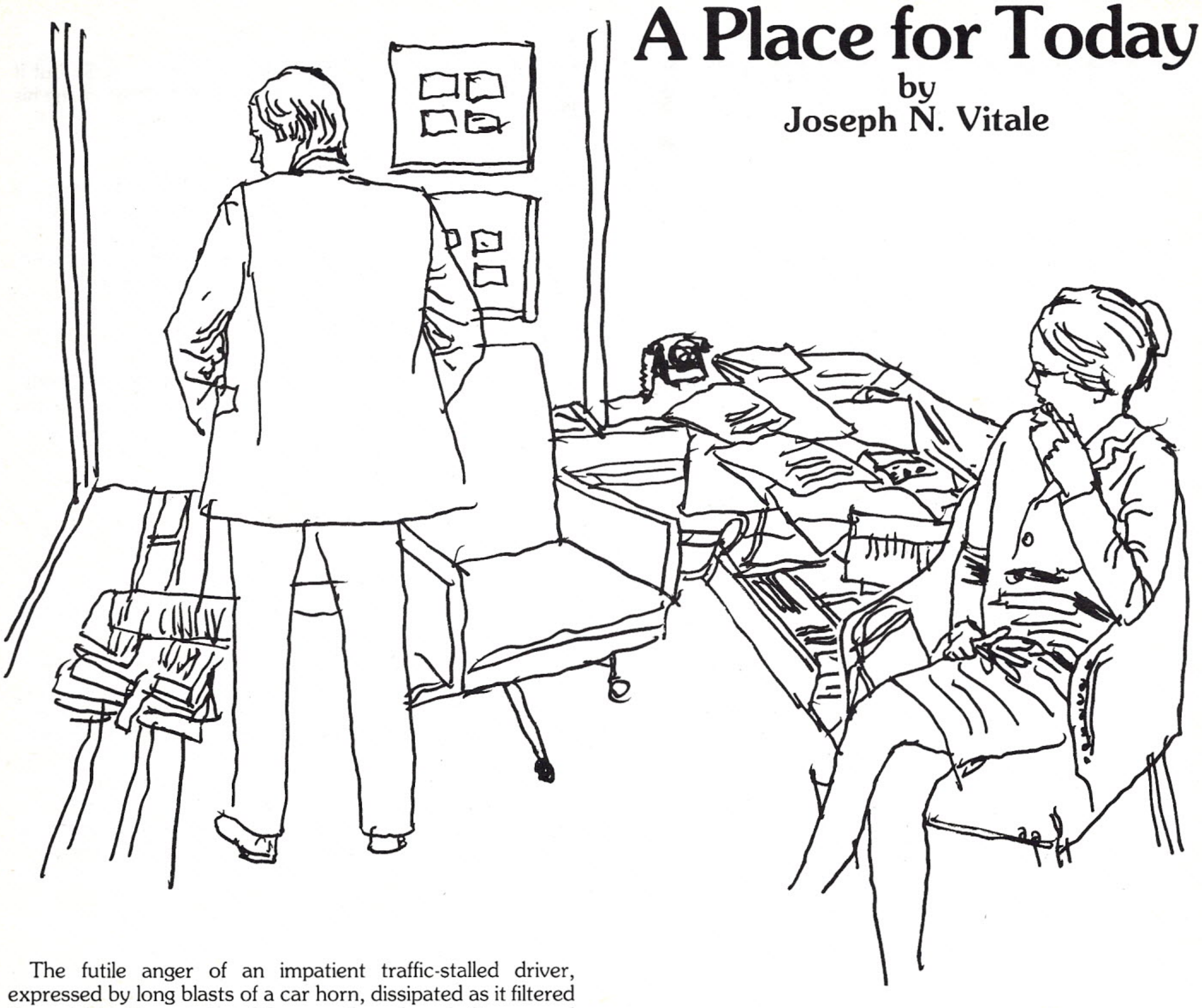
COMPUTER LITERACY QUIZ

1. All computers understand the language BASIC because, as its name implies, it is the most fundamental of computer languages.
2. For any problem within its capability, a computer can always solve it more quickly and cheaper than can be done manually.
3. Example of random access storage devices include:
   1. core and disk
   2. magnetic tape and punch cards
   3. disk and magnetic tape
   4. paper tape and punch cards
   5. all of the above
4. M.I.C.R. stands for magnetic ink character recognition, and is used on bank checks in the United States.
5. Which of the following does not manufacture and sell computers?
   1. Control Data Corporation
   2. IBM
   3. Digital Equipment Corp.
   4. Honeywell
   5. American Telephone and Telegraph Company
6. A typical CAI drill and practice program:
   1. works only when one is teaching elementary arithmetic
   2. aks the student questions and checks his answers
   3. forces all the students to work the same set of problems
   4. allows three incorrect responses before going on to the next problem
   5. none of the above
7. Although learning a machine language is difficult, once one has mastered it, he can write programs that will be understood by any machine.
8. The best computer programs for playing chess and checkers are based upon having the computer memorize tens of thousands of board positions (i.e. rote memory).
9. It is now possible to manufacture a single large-scale integrated circuit, called a chip, which contains all of the circuitry for a CPU.
10. The concept and use of punched cards was developed:
    1. before 1900
    2. about 1920
    3. about 1940
    4. about 1960
11. PLATO is an educational computer system which uses a gas plasma display terminal.
12. In the early days of computers, all programming was done:
    1. in FORTRAN
    2. in BASIC
    3. in machine language
    4. in UNIVAC
13. The Turing "Imitation Game":
    1. has a computer imitate a business environment to train executives in decision-making.
    2. has a person imitate a computer to find program errors.
    3. has a computer simulate a complex situation providing a detailed study of alternative effects.
    4. has a computer pretend to be human, demonstrating artificial intelligence.
14. The science of control and feedback theory is called cybernetics, and Norbert Weiner contributed a lot to this area.
15. One threat to privacy comes from the willingness of most people to provide information about themselves voluntarily.
16. Which of the following is a characteristic of a problem which is well-suited to solution by the computer?
    1. Problem solution involves value judgments
    2. All necessary decisions are quantifiable
    3. The problem is ill-defined
    4. The solution to the problem is needed only one time
17. The largest user of computers in the U. S. Government is:
    1. The Internal Revenue Service
    2. The Census Bureau
    3. The military
    4. Congress
    5. None of these.
18. When one is buying a computer system, he might purchase hardware and software from two different companies.
19. Magnetic tape is an effective medium in operations requiring frequent access to data on a random basis.
20. Very large computer programs are apt to contain undetected errors even after the programs have been used for several years.
21. NCIC is a method whereby checks printed with a special ink can be machine read.
22. By 1950 about 1000 electronic digital computers had been manufactured and placed into service.
23. The fastest core memories have retrieval times of about one millisecond.
24. Using an 8 bit code (such as on magnetic tape), how many different characters can be represented?
    1. 8
    2. 16
    3. 32
    4. 256
    5. 512
25. A computer's memory can think about and solve a problem much in the same way as a person's brain works on a problem.
26. A major problem with computerized data banks is guarding against erroneous data getting into the system.
27. A disadvantage of punched card machines is that the speed of processing is limited by the movement of mechanical parts and devices.
28. Which of the following is not an example of the administrative application of computers in education?
    1. Payroll
    2. Scheduling
    3. Student records
    4. Computer-assisted instruction
29. Why do computer scientists write computer programs to play games?
    1. Computer scientists have lots of fun doing this.
    2. To communicate the ability of the computer.
    3. To study the nature of problem solving.
    4. All of the above.
    5. None of the above.
30. Computer costs (measured in terms of computations per dollar) have leveled off in the last five years.

Answers on page 81.

A Place for Today

by
Joseph N. Vitale



The futile anger of an impatient traffic-stalled driver, expressed by long blasts of a car horn, dissipated as it filtered through his office window. It became the only melodious sound he had heard all day. Up until then he sat, mind and body bathing in dreariness, staring at the rain-soaked window and listening to an almost-cyclic splatter. The wind seemed to drive periodic herds of rain into that side of the building. Blow, you bastard, he thought, it's music to my ears! As if commanded, the car horn responded even more irately.

Christ, it's dark outside! What time was it? As he turned toward the clock on the wall opposite the window, his eyes scanned a variety of computer-generated art forms: Miss September Playmate of last year with heavy typescript in the right spots; a yellowing Santa Claus sleigh led by only six reindeer above which were the letters *Merry Xmas and a Happy*, the rest had been torn off some time ago; several neatly-drawn trigonometric curves among others. It was only after he noted the time at 3:15 PM that he let out a groan from the pain that originated in his neck. Too young to get a stiff neck on a rainy day.

He looked down at his desk; papers full of mathematical equations, large sheets covered with logical flow diagrams, assorted printouts and decks of punched cards. What did it mean? He studied his latest project; a speedier technique by which the computer could approximate the numerical solution to a set of coupled differential equations. What the hell did it all mean?

The rain seemed to fade and in his mind the quietness of the empty office amplified until the silence was olive-oil pure. Then he was no longer alone! They came again! Large faceless bodyless heads with long hair, medium-length beards covering high-necked collars, through his office doorway, across his desk and out his window, each with a name lettered over an empty face: Leibnitz, Gauss, LaPlace — on they came — LeGendre, Fibonacci, great mathematicians all! And again, also, he realized that they had sat in such offices, and on such dismal days as this, most likely, established much of the theory he was using. And he? He had contributed nothing today, a void which was made even more monumental by the fact that he had five million dollars worth of computer at his disposal. The rains came again and he despaired. I am nothing!

His depression settled on a buzzing sound deep within him. He decided he was hearing himself think. Christ, he thought, not only can't I get my mind together, but I've got to listen to myself producing nothing. But wait! It was outside of himself. The intercom on his desk was buzzing away! He picked it up and his secretary announced the arrival of his 3:30 PM appointment with Miss Ann Coyle.

The young woman sat across the desk from him, her legs crossed, dress raised over the upper knee to an inquisitive length; her long sandy hair was drawn tightly into a bun high on the back of her head and large thick-rimmed circular glasses covered deep-blue eyes which betrayed an 'I dare you' look.

Not another one! Not today! Not after Gauss, LeGendre and company! He was constantly beseiged by a variety of people who wanted demonstrations of the computer's incredible feats:

tic-tac-toe, horoscopes and various other games ingeniously programmed by people like himself to give the computer a humanistic appearance. What bullshit! Inwardly, he was satisfyingly-proud of the fact that he knew of one binary digit — called a 'bit' in the computer — which, if changed in the computer's central processor from a zero to a one, would drive the machine stark raving mad; a berserk multimillion dollar totality of illogic, destroying itself ad infinitum or until someone discovered and pulled the plug.

"I know nothing about computers," she said, her voice humble but firm. So what else is new, he thought. "My primary interest is the philosophy of art," she continued. The what of art . . . the philosophy of art! What the hell?

". . . and I had been directed to you." she finished an explanation, much more confident now. Her skirt rose slightly higher on the knee.

"How do you think I might help you?" he heard himself asking. "Computers have been used to generate artistic pictures using special plotting devices. There are situations, at some museums, where computers have been used to do selective retrieval from files of art objects. For example, given complete computerized collections of all paintings in a particular museum, certain retrieval programs can be used which might list out a total description of those paintings which satisfy chosen sets of properties presented to those programs." He looked up at the crack in the ceiling, almost hoping that water might seep through. He continued, "I suppose there are needs, by various scholars, also, to do statistical analyses by computer on data that physically describe paintings in certain groups . . . like what is the average weight of all paintings in a particular group, or mean canvas areas, or something like that." He stopped to let her absorb his statement.

"But," he went on, in a rapidly tiring voice, "I'm not sure I can see how a computer might be used when dealing with the . . . the philosophy of art." He raised his eyebrows and peered heavily through her glasses. Am I sinking in, he wondered? No, I am not, he answered his own question.

"Please correct me if I say anything that sounds like nonsense," she said. "I have heard that the computer can do . . . uh . . . unbelievable things and I wonder . . ." her voice oozed lazily into silence.

"You wonder what? Let's have it!" Christ, let's have it!

"Well, I wonder if the computer can get two people to talk to each other and discuss their philosophies!" she rambled out the phrases in a quick jerky rhythm.

"Oh, is that all? You don't need a computer for that! What you need is called a telephone!" He was getting somewhat impatient. This girl is either very misguided or very strange.

"No, let me finish." She smiled. "The people I am talking about lived at different times! One is Marcel DuChamp, the French-American artist who died in 1968. The other is Marshall McLuhan, the American art philosopher who is still alive." She spoke much more confidently, now. "You see, it would be intriguing to hear them discuss their philosophies with each other. It would be particularly interesting if the computer could generate a dialogue between them in the form of a two-man play. While . . . although their lives overlapped . . . it's the one thing they never did . . . talk to each other. And yet, it would have been one of the most stimulating discussions in the history of art philosophy." The skirt was definitely up higher than ever now and the eyes twinkled, almost immorally.

He gulped hard. He rose from his desk, walked over to the window and looked outside. It had stopped raining now and it was much lighter. A rainbow was forming to the West. The traffic was moving a lot freer now, too. They went out this window . . . Leibnitz, Fibonacci and the rest. Where did they go? And what did it mean?

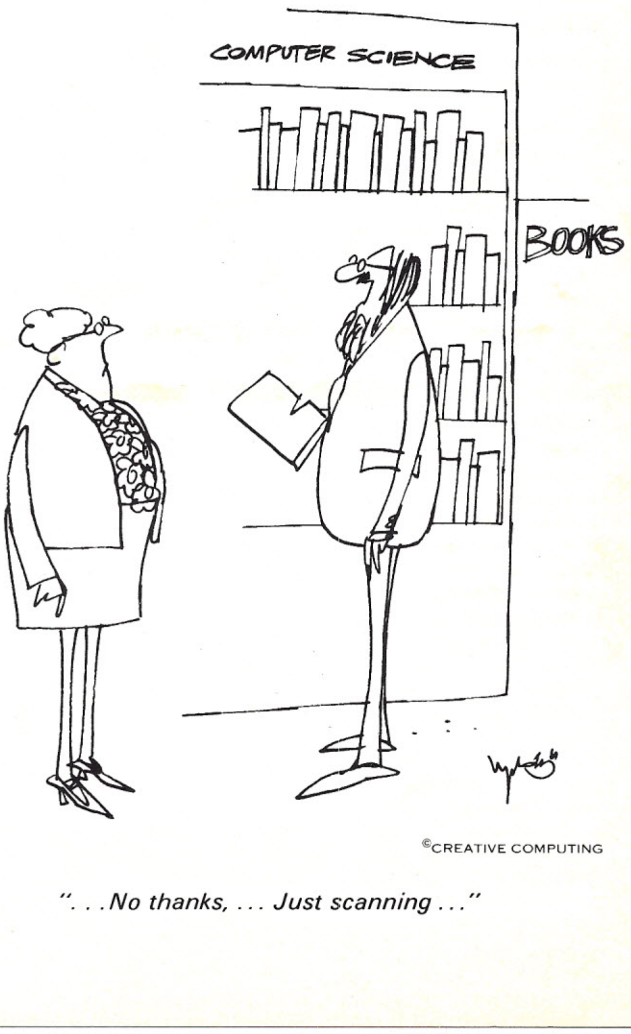
He turned and stared at the back of her head, concentrating on the bun as if it were some crystal ball in which he might see some meaning. Was there any special association between the events of today — the rain, the parade of the faceless ones, this girl with her philosophy of art, DuChamp and McLuhan, the computer and the rainbow? Was there any special association?

Association? Association! The phrase came slowly, but it came, first rolling soundlessly off his tongue, then circling his mind like a stock market ticker-tape. Linear Associative Information Retrieval! He had read of it some months ago in one of the journals; a new theoretical technique which could be used to determine the sameness or differentness of two documents . . . to indicate the snynonymy or contiguity of two documents by using similarity matrices with respect to assigned keywords which described the documents. Weren't the ideas of sameness or differentness when applied to concepts rather than documents a way to distinguish philosophies — basically the writings of the men! Yes, yes! His mind accelerated now, building the basic steps, already seeing mental images of flowcharts, computer programs and, finally, dialogue printing at a computer terminal! Dialogue between DuChamp and McLuhan!

And wait! Why stop at DuChamp and McLuhan? Certainly, if the computer could be made to generate discussion between them, then the possibilities were endless. What about Adolph Hitler and Karl Marx? Or William Shakespeare and William Faulkner? Or any two people who had published a great deal? Or any three people? Unbelievable! But possible! Round table discussions between any of the eminent people in the history of the world!

A large bodyless head came through the doorway of his office, across his desk and out the window. Only this time it was not faceless!

"Miss Coyle," he said, "could you bring me some literature which contains a representative sample of the philosophies of DuChamp and McLuhan? I think . . . I think I might be able to help you."



©CREATIVE COMPUTING

". . . No thanks, . . . Just scanning . . ."

THE REACTIVE ENGINE PAPER

BY TERRY WINOGRAD

This paper, written in October of 1974, originally appeared as a file on the computer at the Artificial Intelligence Laboratory at Stanford University where Terry Winograd is currently leading courses in computer science, linguistics, and the social implications of computing. At the time of writing the people at SAIL were discussing how to design their new timesharing system. Another file was also maintained on the system in which comments, suggestions and objections could be stored. It is an argument for personal computing — not an essay, but a sort of cybernated commentary.

—Marc Le Brun

NOTE:
This is written in an attempt to provoke discussion, so it may overstate, exaggerate, etc. It is written from the viewpoint of the Ivory (silicon?) tower, so the issue is "What is possible?", not "What is now practical?" After all, working at the AI lab you would never realize that there are still people in the world (even at Stanford!!) who talk to computers by cutting little holes in pieces of cardboard.

WHY TIME-SHARING IS OBSOLETE:

Time-sharing is obsolete because it is based on the assumption that a person interacting with a computer large enough to do serious work cannot make good use of its computing power except during a fraction of the time.

GEDANKEN EXPERIMENT:

Imagine that by a decree of God (or ARPA or whoever) we were only allowed to run one user at a time on the PDP-10. All the systems efforts would be directed to making each person's time on the computer as profitable as possible. What kind of system would you organize?

REACTIVE SYSTEMS:

I will borrow a term from Alan Kay (from whom I have also absorbed many of these ideas) to contrast our current "interactive" system with an imagined "reactive" system. A reactive system contains a full-scale processor whose only job is to cater to a single user. It can therefore afford to do relatively large amounts of processing when he or she is doing the simplest of tasks, like editing or giving monitor commands. It can maintain complex reactive graphic displays in real time (e.g. the multiple windows in the current SMALLTALK system).[1]

CLAIM:
The advantages of a reactive system over current interactive systems will be as large as the advantages of interactive over batch.

NATURAL COMMUNICATION:

One of the main advantages of a reactive system is that it can afford to do extensive processing to figure out what the user wants to do, based on both what he or she says and what the current context is. This is the main feature of natural language as a communication system — it is designed (evolved) with the assumption that the hearer will always make use of context and a shared base of knowledge in his interpretation. People are much more efficient at communicating in this style, regardless of whether it is in actual "natural language" or in some artificial language. There are lots of bits and pieces of this in current systems — everything from default file extensions to command completion to the rather extensive set of facilities (like spelling correction) in INTERLISP.[2] On current interactive systems there is a strong tendency to avoid these because they involve running a higher-level program to interpret inputs. With a reactive system, the bottleneck is the user's typing speed (although we could even imagine some sort of simple voice inputs if we pushed this idea far enough), so we can afford to do lots of processing.

REMEMBERING AND DEDUCING:

Often the problem isn't that we want to specify a command in a way which needs a smart processor to understand what it is, but in a way which takes a smart processor to do it. If we were talking at the command level to a program with even limited deductive capabilities (the kinds now put into robotics programs) it would free us from much of the tedium of converting our desires into "machine code." Why shouldn't we be able to say in some language "put the current who display in a file" and let the system worry about the details of just what needs to be run.

The whole range of things like UNDO, REDO, etc., which are creeping into languages like INTERLISP could be extended to cover every aspect of our communication with the computer — it could remember the context of what we have been doing, and make the necessary deductions to do rather sophisticated things (e.g., redoing some sequence of actions, but tailoring it to apply to a different kind of data object).

EDITING: **Maxim: People Never Edit Character Strings.**

Whenever I am editing, I am editing a document, a message, a program, or some other structure about which I know much more than the simple sequence of characters. The reactive system should always be dealing with my editing in this way. Formatting systems for documents should be interactive and incremental — what I see on my screen should always be what I would see on the Xerox Graphics Printer when I put it out. This includes fonts, justification, diagram placement, etc. etc. Of course the program should be able to reconfigure and modify this in a whole variety of ways, but I should always be working with a document, not a source file (or even a screen-editor!) When I put in a new word or line, things should move to make room for it. Things like spelling-checking could be done incrementally, looking up each word as it is typed in, then when I ask for it, interactively pointing out those not in the dictionary (e.g. by flashing them) and letting me make changes. For programs, the editor should be part of an integrated programming and debugging system (as in INTERLISP), not a separate program at all.

PROGRAMMING AND DEBUGGING:

We have glimpses of integrated systems in SMALLTALK, INTERLISP, ECL[3] and Swinehart's thesis[4] describing a SAIL-based system. Often these are forced into horrible compromises by the fact that they are walking the line between wanting the user to have the full power a program can offer at each step of the way, while having to run them on time-sharing systems which are based on the assumption that you really only want to process part of the time. They fall far short of putting in the kind of integrated knowledge-base and deductive programs which would really allow the system to act as a programming assistant. They do not make full use of the possibilities for incremental compiling (coupled with the editor) which can give the user the feeling of always working with an interpreted system, while actually having the efficiency of careful compiling. They are only beginning to make use of the kind of graphic interactions which can greatly broaden the programmer program bandwidth. I have

Reprinted with permission of the author. Copyright 1975 by the *CoEvolution Quarterly*, Sausalito, CA 94965.

ranted about this elsewhere[5] so will say no more here, except to note that this is probably the most important area where reactive systems will change our concept of programming.

IS IT WORTH IT?

The same could be (has been and still is) asked about interactive computing. Ask the programmer who has become accustomed to an interactive debugging system and then has to go back to octal (or hexadecimal) dumps. In the short run (and we are always in the short run) this is a difficult question, and I will take the escape given above — it is certainly worth it if our research grants will support it. In the long run I don't think there is any doubt. The costs of processing, memory, etc. are going down and although there may be ultimate limits, we are still far from them. Anything which trades off increased processing for increased ease of computer use by people will be eventually justifiable on practical grounds. Twenty years from now this question will be as outdated as it would be to ask now whether it is really worth having the computer go to the work of accepting programs in symbolic form, since people are perfectly capable of converting them to binary numbers and entering them that way.

CAN IT BE DONE WITH CURRENT TECHNOLOGY?

Researchers at a number of places (including XEROX and MIT-AI) are currently designing and building "personal computers" which have a processing power far beyond standard minis or "smart terminals." In fact many of the ideas above are being implemented at Xerox. By connecting these into a simple network with shared facilities for things like specialized I-O (XGP, A-D and D-A, ARPANET, etc.) and large file storage (allowing file sharing), these computers could currently provide everything we get from a time-sharing system, while opening the way to all of the advantages of reactive systems. At the moment, the cost per user (even if they were commercially available — which they aren't) would be significantly greater than an improved this-generation time sharing system of the kind being planned at SAIL. However the picture is not that clear if we look ahead the 3-5 years necessary to actually get systems running. Careful thought needs to be given to realistically assessing what hardware will (or could if we put effort into it) be available by then, and how much it will really cost.

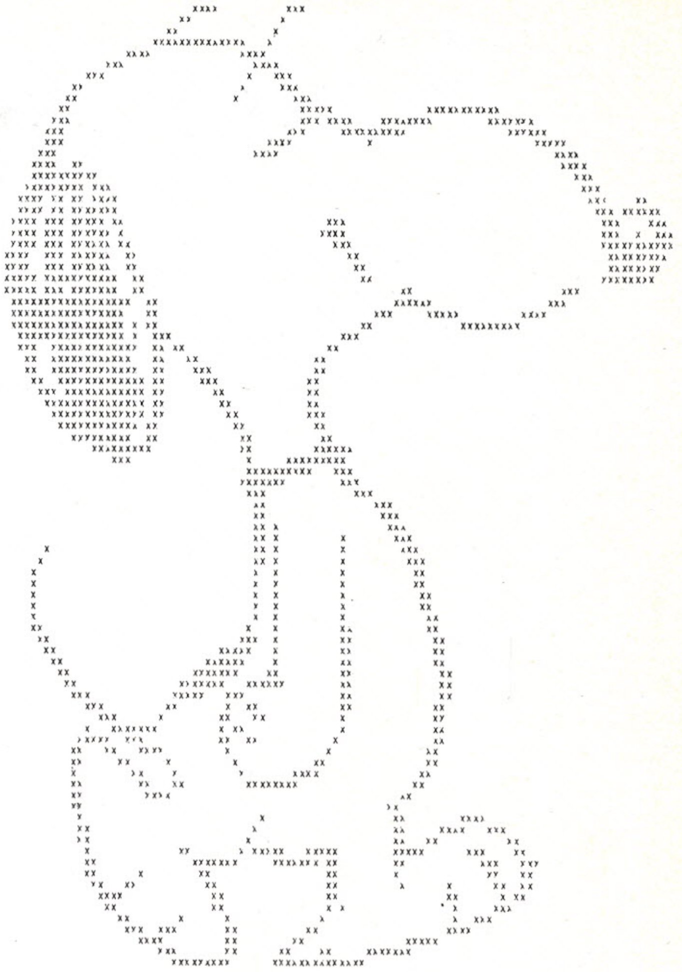
WHY SHOULD WE CARE?

This paper was circulated only partially because there is a general interest in talking about systems. There are particular reasons to think seriously about these issues now. AI researchers (particularly John McCarthy) have always been at the cutting edge of new systems ideas (like time-sharing, list processing, etc.). In this case I think we have even more to offer, since many of the techniques which are needed to build reactive systems are "AI techniques" involving knowledge bases, deduction, etc. We have the option of taking the viewpoint of "wiping the slate clean," thinking in terms of what we would really like to do with the new technologies, rather than incrementally building on the ideas and equipment now available. Even if the appropriate personal computers were not available right now, much of the design and debugging could be done on current systems if we had a clear idea of what we expected from the hardware.

PUNCH LINE: I believe that reactive systems are "where it will be" a few years from now, and we have the choice of leading or waiting to follow. ■

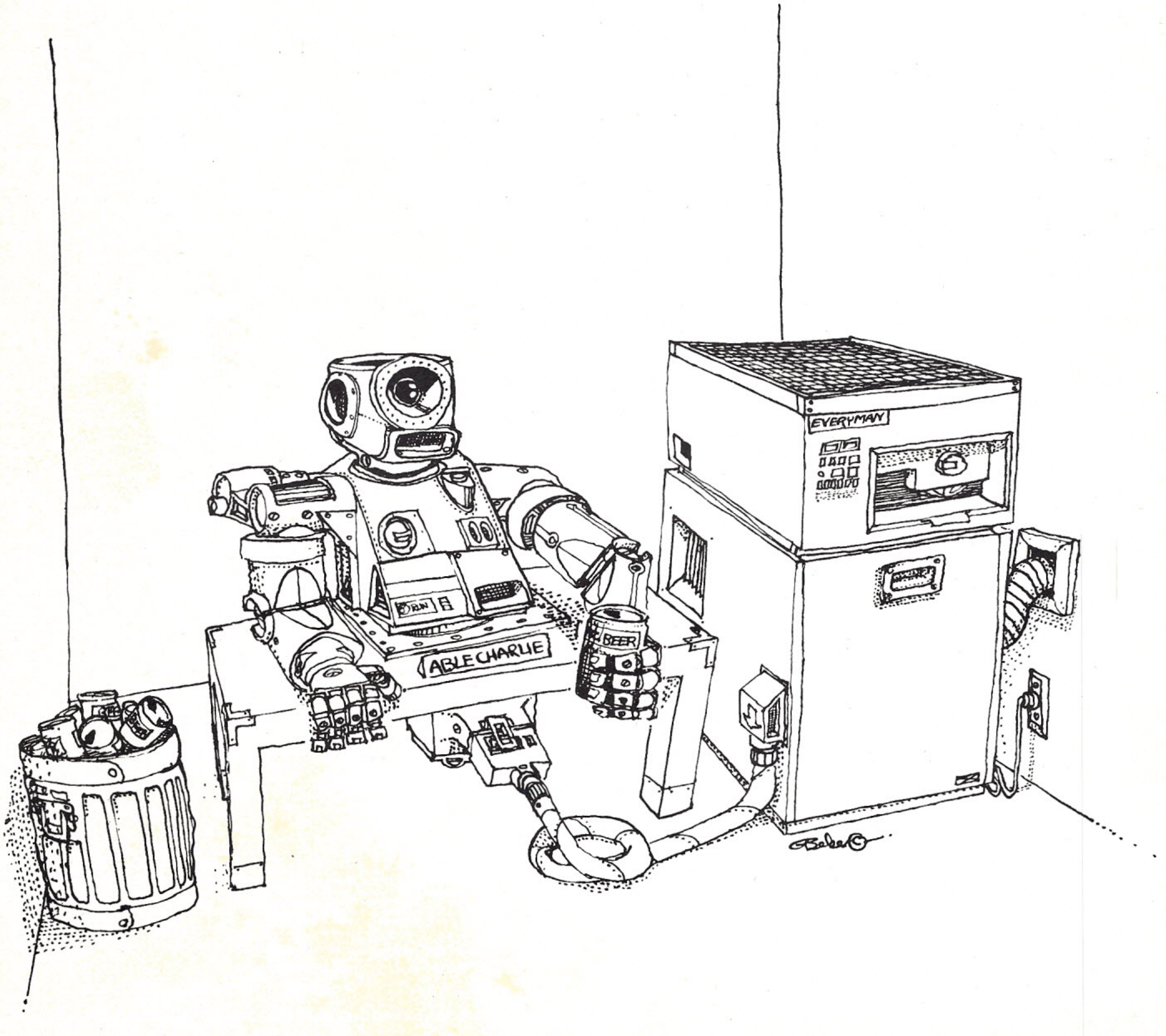
REFERENCES

1. **Personal Computing by Alan Kay, Learning Group, Xerox Palo Alto Research Center and A Personal Computer for Children of All Ages by Alan Kay, ACM National Conference, August 1975.**
2. **INTERLISP Reference Manual by Warren Teitelman, Xerox Palo Alto Research Center.**
3. **The Treatment of Data Types in ELI by Ben Wegbreit, Communications of the ACM 17, 5 May 1974.**
4. **COPILOT: A Multiple Process Approach to Interactive Programming Systems by Daniel Carl Swinehart, Stanford Artificial Intelligence Laboratory AIM 230.**
5. **Breaking the Complexity Barrier (again) by Terry Winograd, ACM Sigplan Notices 10:1 January 1975.**



A Day in the Life of Able Charlie

by Frederik Pohl



The time was 0900:00 A.M., and Charlie woke up.

The first thing he had to do was to find out who he was that day, and so he explored his memory. He discovered that he was a white male American, thirty-two years old, married, employed in the sales department of a public utility company. He had two children, a boy and a girl. He had made $17,400 in the year just past, and if it hadn't been for Harriet's part-time teaching salary he didn't know how they would have managed. He still owed over $19,000 on their $38,000 house, $1900 on the car and nearly a thousand on the loan for modernizing the kitchen they had taken out two years before. Moreover, his daughter, Florence, had unfortunately inherited his bite, and so the orthodontist was going to cost him fifteen hundred dollars very soon. Charlie discovered that many of his thoughts were of money.

However, his memory contained many other things. He became aware that he was a fan of the Los Angeles Dodgers, and that he had volunteered as a Little League coach against the day when his four-year-old son, Chuck, was old enough to play. Charlie remembered that he was inclined to favor Chuck over the girl. It was curious that he could not remember what color Chuck's hair was, or whether Florence was doing well in school, but Charlie didn't realize that it was curious and so he continued to explore his memory.

He was a heavy smoker, drank a can of beer now and then, especially in hot weather, but didn't go much for the hard stuff. Although he liked looking at other women, he did not go beyond looking. Although he enjoyed a game of poker twice a month, he did not care to gamble heavy stakes. He drove a small foreign car (it was not clear whether it was a Datsun, a VW or a Fiat), on which he got 24.7 miles to the gallon in everyday use and nearly 29 miles a gallon the road. (He did not know what color it was. It did not occur to him to wonder why.) Charlie remembered that he was active in his party's politics (he did not know whether it was Democrat or Republican) and that he thought the mayor of his town was a crook. But he could not have said the mayor's name.

All these things about himself Charlie apprehended in a very short time indeed. He then spent somewhat longer remembering what brand of cigarettes he smoked, where he bought them, what had happened when he tried to give them up (his wife complained of his short temper and begged him to start again) and what other brands he had tried. He rehearsed the services offered by his neighborhood filling station, and what he looked for when he needed gas on the road; what kind of Scotch impressed him when he was offered it at a friend's home; and why he had decided against switching from lather to an electric razor. Charlie inventoried every purchase he and his family had made for the past year, swiftly and without error. He recalled what TV programs he watched, what magazines he read and which of the thousands of commercials and advertisements they contained had affected any of the purchases.

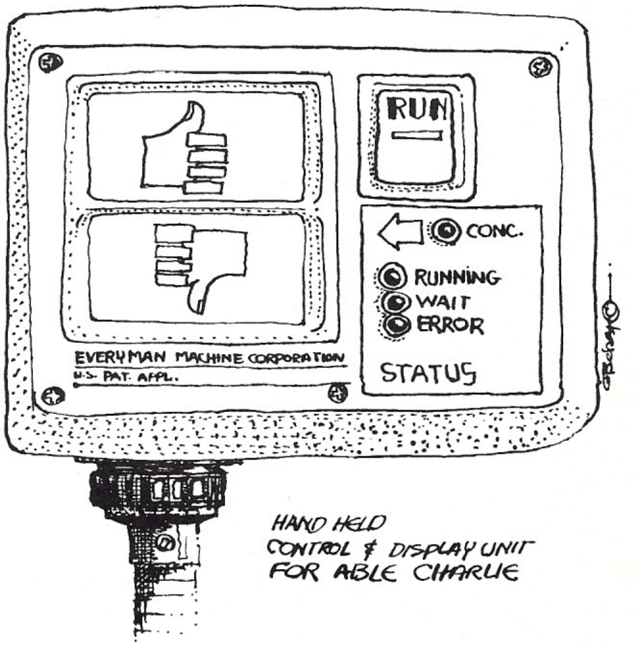
At that point Charlie discovered that he had done everything he was required to do just then. He made a quick parity check on his instructions. When it revealed no gross error or failure on his part, he announced that he was ready for his next task and waited in standby mode for orders.

He waited what was for Charlie a very, very long time. All of this had taken Charlie a period measured only in fractions of a second. Now he rested, neither wondering nor moving, for a stimulus to further action. Without it he would do nothing, ever. He was not impatient. He knew what "patience" was in conceptual terms — he could relate it to his memory of himself waiting without "patience" for a traffic light to change — but it did not occur to him to feel that way now.

At 0901:30, give or take a few seconds, a young woman in a light gray dress, carrying a container of coffee, set the coffee down on her desk and seated herself before a large typewriter. She had heard the bell that announced Charlie was ready more than a minute before, but she was not quite ready for Charlie. She typed several rows of characters, checked them over, took a sip of her coffee and stood up.

She glanced at the various lights and dials on Charlie's front panel, saw nothing to cause concern. Her typewriter had produced not only the visible row of characters on the sheet of paper it held but, on a spool connected electrically to the keys, a strip of magnetic tape. She snipped a four-foot length of it free, taped it to another reel, rewound it and fed it into a scanning device. She removed the rubber band from a packet of perforated cards and dropped them into a hopper.

Then she pressed a button. Rubber-tipped fingers dealt the cards into sorting bins where, one by one, they were taken up again and read, like the music roll of an old player piano. The tape reel slid past its scanning head on a cushion of air and disappeared. The time was 0901:55.



Charlie began work — not at 0901:55, exactly, but at a time so near to it that the difference was measurable only in picoseconds.

His first problem, he was informed, had to do with cigarette package designs. He waited while the cards on that subject were scanned. There were forty-one alternate designs, and they were presented to him in pairs. First he was offered Package One and Package Two simultaneously; he compared them, made a value judgment based on what he knew of his own buying habits and preferences and stated his preference. Then Package One and Package Three were offered to him, then Package One and Package Four, and so on until Package One had been compared with each of the others. Then he was offered Package Two with Package Three, Package Two with Package Four; and on and on until each prospective design on the list had been compared with each other. (There were 861 combinations in all, taken two at a time.)

At that point Charlie went into a sort of reverie while another part of his mind — it could have been called his "subconscious" — tabulated the results of his cross-pairing and established an order of preference. He wrote down, in order, the ten package designs he had most favored. He wrote it in the form of impulses recorded on a magnetic tape (this caused a reel by the desk of the girl in gray to spin rapidly for a moment, which she noticed out of the corner of her eye). Then he hummed for a moment, waiting for the card reader to allow him to begin his next task.

Each of Charlie's value decisions had taken him only about four nanoseconds, but the evaluation and read-out were much slower. It took him considerably longer to announce his results than to arrive at them, and so it was 0902:45 before he began his next job.

The next assignment was to assess the merits of some proposed shaving-cream formulations.

Here the task was considerably more difficult, for several reasons. The first part of his task was to rank his preferences among the fifty-five formulations as to their odors, textures and visual appearances, each in combination with the other. Charlie did not, in fact, realize quite how difficult it was, since he had no idea that he possessed neither smell nor vision, and touch only in the sense that certain of his members were capable of probing a card or tape for punched holes. He then had to evaluate some 24 shapes and weights of pressure canisters in relation to each sort of lather. Here too, Charlie was unaware of his lacks. In fact he did not have thumb and fingers; the "grasp" and "weight" and "feel" of the canisters in his "hand" was in fact only a locating of certain binary statistics within the parameters of certain other quantities that were a part of his memory. In order for Charlie to be able to express an opinion on any of the matters on which his verdict was sought many subterfuges had been devised by the programmers on the staff of the advertising agency that owned Charlie. They materially prolonged the time for each comparison. However, he was in no way concerned by this. He did what he had always done. He did the task that was assigned to him, and when it was done he looked for, and did, the task that was next.

In all of the hour and forty-odd minutes in which Charlie, husband of Harriet, father of Florence and Chuck, searched his responses to a wide range of offerings, he performed something over five thousand million separate operations, including parity checks and internal verifications. He faithfully reflected the customs and tastes of the average of a sample of some four million American males as they pertained to the purchase of tobacco, beer, gasoline, automotive accessories, soft drinks, airline tickets, motion-picture admissions, sporting goods, hi-fi equipment, toilet articles and power tools. When his final magnetic report was on the tape he signaled by ringing a bell. That was the end of Charlie's working day. In a sense it was the end of his life.

The girl in the light-gray dress was in the assistant division chief's office when Charlie's bell rang, and so she didn't react at once. Charlie waited like a man on a benzedrine high, his mind clear and capable, but disengaged. It was nearly 1100 when the girl got back to her desk.

She took the spool of tape that held all his opinions and threaded it into a printer, where it began typing out plain copy at a rate of 350 words a minute. She replaced it with a blank spool, consulted her work sheet and began to change Charlie with switch, with patch-cord and with dial.

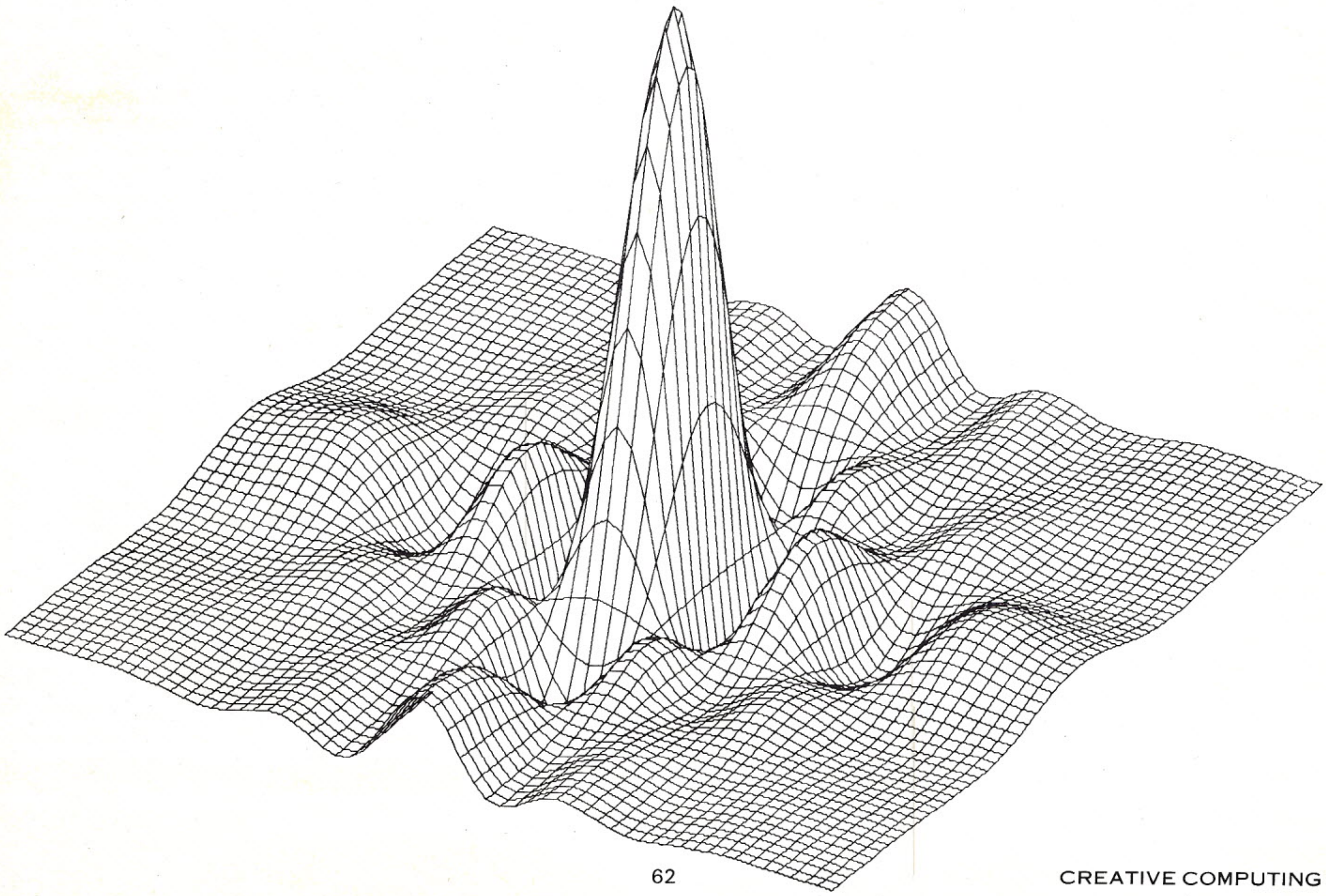
As she worked whole banks of memories dropped out of circuit. Chuck and Florence fell out of his personality without leaving a mark. His wife disappeared, his house, his car; the Los Angeles Dodgers went, with the Little League and the dunning letters from the bank.

She then checked the programming sheet and, following its instructions, selected new personality ingredients for Charlie: an economic level, an age, a set of buying habits, a profile of interests. She began to charge Able Charlie with the sum of these habits and biases. He was not yet aware of what he was, since he had not yet received the command to learn himself. For that matter, he was no longer "he". Now Able Charlie was a teen-age girl, her principal interests cosmetics, soft drinks, clothes, records and boys.

When all the patches were complete and the new tapes were ready to roll, the girl in the gray dress double-checked, and pressed the "execute" button. Able Charlie, AC-770, began to take up his — her — its new life.

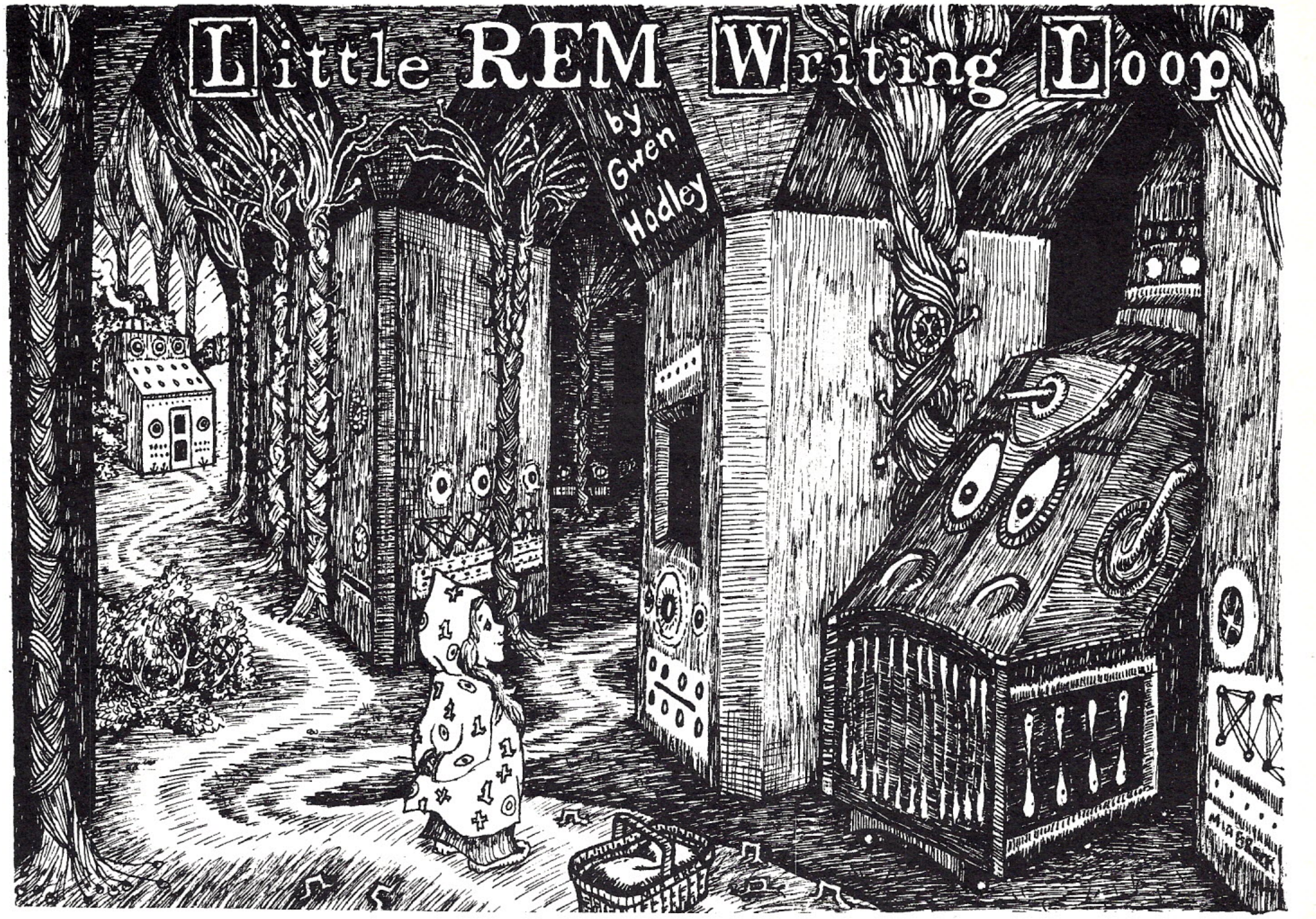
The girl in the gray dress idly examined the polish on her nails. Her mind was not far from stand-by mode, either; until the first read-out came, or a trouble signal, she had nothing to do but wait for lunch.

Inside the AC-77 Charlie, or Charlotte, was swiftly sniffing colognes whose fragrance was only the simulation of magnetic patterns on iron-oxide tape and comparing shades of lipstick whose colors were only a point on a hypothetical scale. The girl programmer was comparing colors, too. She wished idly that she had a friend to chat with — Rose Pink, after all? Or Catalina Coral? — but when she thought she heard a low contralto sigh she dismissed it at once. She knew that she was alone.



Little REM Writing Loop

by Gwen Hadley



Once upon an operating system, in a cassette on the 12-edge of the forest, there resided a binary decimal digit. She was counted by her REM statements whenever she went looping, and so she was labeled Little REM Writing Loop.

One day, Little REM Writing Loop's matrix called her from her subroutine. "Will you convey this disk pack to your nano-matrix?" she printed. "She is off-line, today, and I am processed about her. Her subsystem has been listing as of late, and I can only conclude that she is headed for third storage."

"Oh, to be positive, Matrix!" was Little REM Writing Loop's voice-response; for she felt right justified whenever she looped down to her nano-matrix's address.

"Be deterministic as you thread your way through the forest," her matrix warned in a Guarded Command. "There could be an optical scanner INFILE. If he monitors this disk pack, he will signal that your nano-matrix is an ALGOLic, when the truth is that she is bugged only by a post condition." She structured Little REM Writing Loop's parentheses over her Is. "You wouldn't want that to be printed out, would you, Little REM Writing Loop?"

"Oh, that wouldn't mark-sense!" returned Little REM Writing Loop. "There are only APLs in this disk pack!"

"Then, BEGIN," instructed her matrix. "I hope your character recognition is Greater than mine."

So Little REM Writing Loop SKIPped out into the forest along the outer loop. In spite of her matrix's Implied Dos, she STOPped to pick some square roots in an address field along the slack path. "This isn't a TAN function," she theorized, "but there is an unambiguous array of hybrids here, and I shall gather them for my dear nano-matrix."

She was picking macro facsimiles when suddenly she encountered an invoice. "Why are you indexing out here among the trees, Little REM Writing Loop?" modulated the invoice.

Little REM Writing Loop raised her (I)s. Dimensioned nearby was a blinking cursor! An optical scanner! A real negative number! Little REM Writing Loop's intuition warned her against time-sharing with this syntactic character, but he was JOSSing. He couldn't possibly have any logical design on her!

"I'm looping to my nano-matrix's threshold with this 8-bit disk pack," she returned. "Nano-matrix is off-line, today."

"Oh," replied the optical scanner. "Where does your nano-matrix reside?"

"She resides in the discrete structure beside the CODASYL," returned Little REM Writing Loop.

Little REM Writing Loop segmented herself from the optical scanner and continued to her nano-matrix's address.

The blinking cursor waited till she had gone into her subroutine loop. Then, he SKIPped through the forest and soon arrived at the nano-matrix's discrete structure. He drummed on the file.

"Who is INFILE?" called Nano-matrix.

"It's Little REM Writing Loop," returned the optical scanner with binary invoice.

"Why, just punch the key and $ENTRY," replied Nano-matrix.

When Nano-matrix recognized the blinking cursor, she vectored and sequenced out of access. Through the network she listed, with the optical scanner on her track.

Then, he STOPped. Instead of auditing her, he SEARCHed through her master file until he found Nano-matrix's most exponential macro. He assumed it and hopped into Nano-matrix's source deck.

Soon, there came a magnetic drum at the file.

"Who is INFILE?" called the optical scanner invoicing like Nano-matrix.

"It's Little REM Writing Loop, Nano-matrix," replied Little REM Writing Loop.

"Just punch the key and $ENTRY," replied the blinking cursor.

Little REM Writing Loop OPENed the file and $ENTRYed into Nano-matrix's structure. She STOPped, and her (I)s left their data base when she looked the optical scanner interface as he multiplexed in her nano-matrix's source deck.

"FORTRAN! WATFOR! Nano-matrix, what amBIGuous (I)s you have!"

"The easier you are initialized, my dear," returned the optical scanner.

"And, Nano-matrix," stated Little REM Writing Loop, "what three-dimensional antennae you have!"

"The better to audit you with, my dear," returned the blinking cursor . . .

"But, Nano-matrix," declared Little REM Writing Loop, "what sharp brackets you have!"

"The better to byte you with," argued the optical scanner. And he LEAPed out of the source deck.

Suddenly, there was a transfer of control. Little REM Writing Loop's Nano-matrix came looping in, and with her, a member of the JCL.

"There he is," cried the nano-matrix, pointing to the optical scanner.

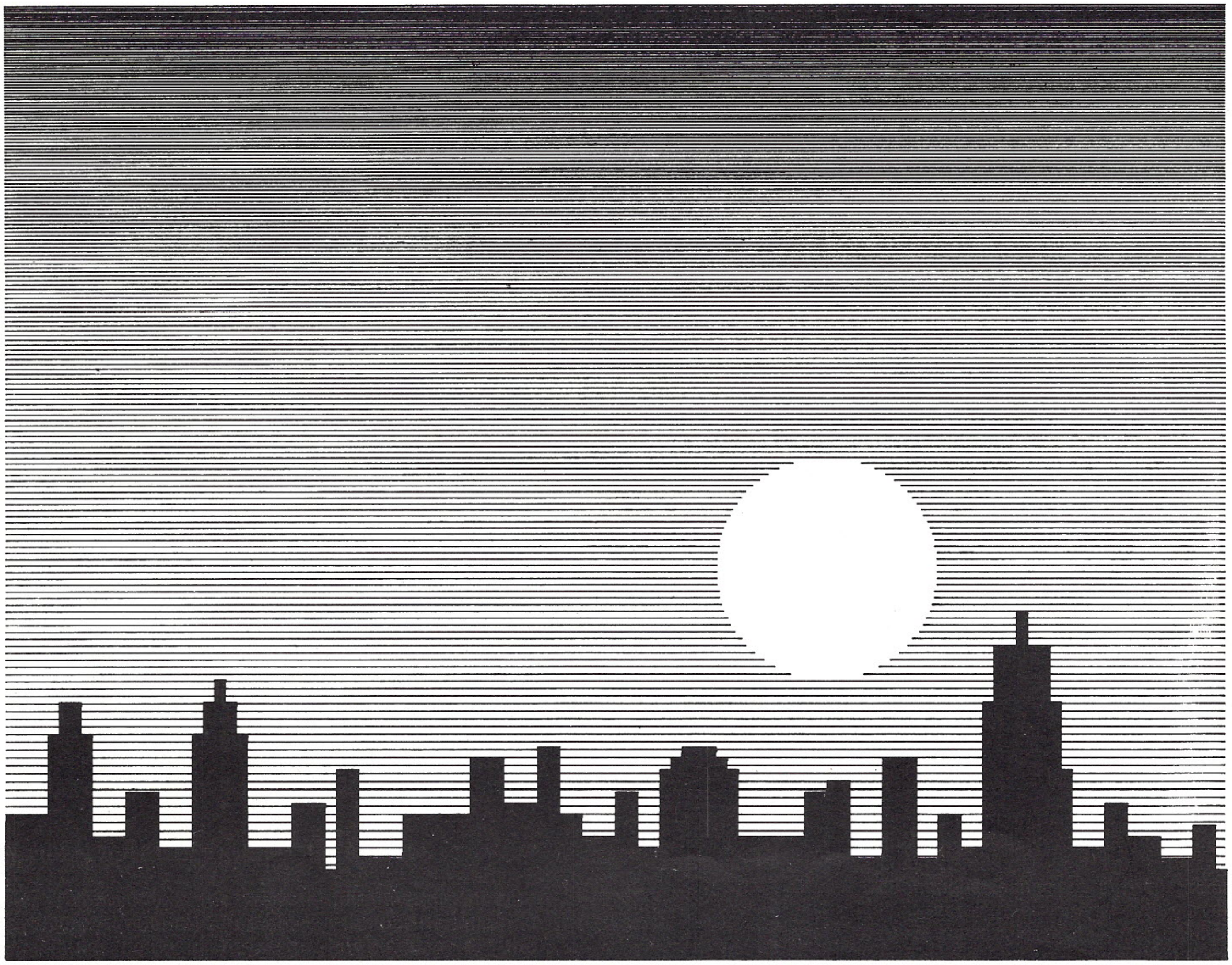
The blinking cursor truncated his stochastic response when he spied the JCL. He indexed around Nano-matrix's decision table and around her analog and took direct access to the outer loop with the JCL right behind him. The JCL loaded a disk cartridge and blanked the optical scanner with a laser beam.

Little Rem Writing Loop backuped her Nano-matrix and helped her into her source deck, for Nano-matrix was at an odd parity, inplemented as she was by an advanced integer problem. "I have been out of SORT, in virtual storage lately," she told Little REM Writing Loop, "but LET us sample the APLs."

So Little REM Writing Loop and her Nano-matrix sampled the APLs until the disk pack was empty and Nano-Matrix's core storage was loaded. And they were never interpolated by the optical scanner again.

"There is a Hierarchy than he," declared Nano-matrix, "and everyone is right justified in the output."

ENDOFILE



"Moonrise Over the City" by Kerry (Bit-Mangler) Jones

10 GOOD REASONS WHY COMPUTERS CAN . . .

A computer can do more work than
a man.
One reason that's little known
Is that it never has to stop
To answer the telephone.

A computer can do more work than
a man.
One more way to explain
Is that it doesn't stop it's work
To argue and complain.

A computer can do more work than
a man.
Because it never takes
Those dawdling, lengthy lapses
That we call coffee breaks.

A computer can do more work than
a man.
And it's easy to see why.
It doesn't sit with its chin on its hand
And watch the girls prance by.

A computer can do more work than
a man.
One reason it's such a whiz:
It doesn't buttonhole passersby
To tell them how busy it is.

A computer doesn't take nervous pills
All day at the water fountains,
And wastes no time with molehills
Making them into mountains.

A computer can do more work than
a man.
Because, I have a hunch
It doesn't spend three hours
With a customer at lunch.

A computer can do more work than
a man.
And one good reason I've seen is
It doesn't spend the afternoon
Half-conscious from martinis.

A computer can do more work than
a man.
And partly it's a matter
Of not spending all day angling
For the next job up the ladder.

A computer can do more work than
a man.
Here's a final explanation:
It wastes no time on fears of being
Replaced by Automation.

Author unknown

NEW ELECTRONIC DESK CALCULATOR FROM CHINA

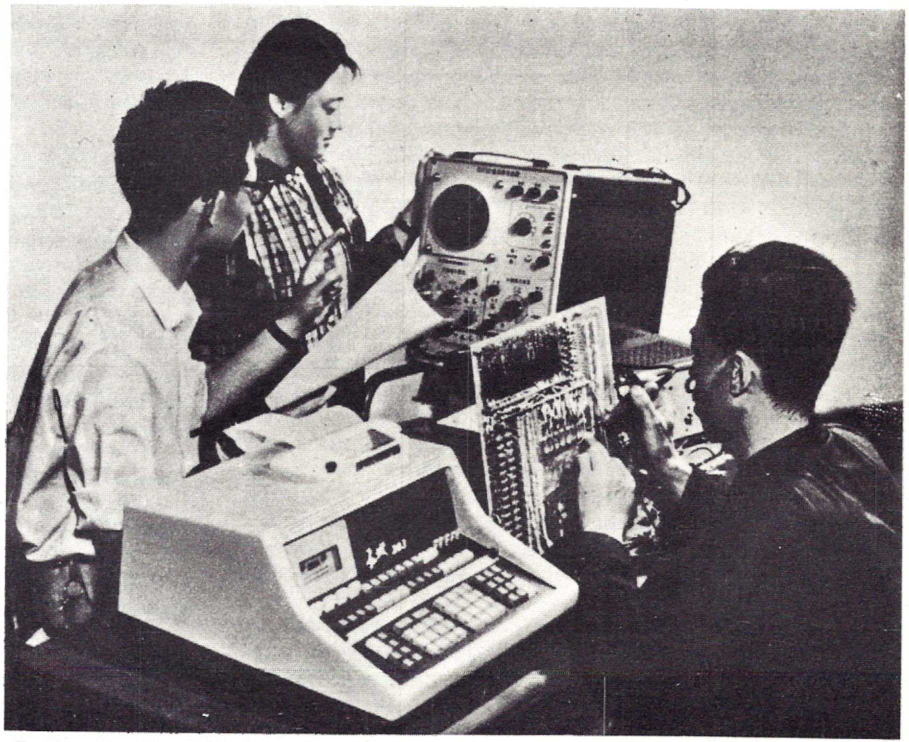
KUNG CHANG

THE Great Wall 203, an advanced type of electronic desk calculator, was trial-produced early this year by a plant under the Institute of Mathematics of the Chinese Academy of Sciences. It is 2.5 times as fast as similar calculators produced abroad, has twice the storage capacity and an expanded machine language. It is also slightly smaller and easier to operate than such models.

The Great Wall 203 is of a type more advanced than ordinary electronic desk calculators. It has more functions, greater storage capacity, higher operating speed and under program control can automatically solve complex problems. Programs can be written, debugged and modified conveniently at the keyboard. Equipped with a printer and a magnetic tape unit, it is a complete, independent small computer system that performs some of the functions of a general-purpose electronic digital computer. Its easier handling and maintenance make it suitable for wide popular use.

Most of the people who designed and built the calculator are young mathematicians. Though they were unfamiliar with electronics and computing and their plant was poorly equipped, they drew encouragement from Chairman Mao's teaching, **"The Chinese people have high aspirations, they have ability, and they will certainly catch up with and surpass advanced world levels in the not too distant future."** After studying a lot of material and critically assimilating the good points of foreign and domestic calculators, they boldly created a design in line with the characteristic of Chinese components. Making full use of collective wisdom, the whole plant made suggestions in the course of trial production.

They designed and built this new advanced electronic desk calculator which uses integrated circuits in one year and five months. Now the Great Wall 203 is undergoing comprehensive testing to perfect it for production and distribution.



Testing the Great Wall 203.

Reprinted from a recent issue of CHINA RECONSTRUCTS, *Peking.*

COMPUTER GENERATED AIDS TO TEACHING GEOMETRIC CONCEPTS

by
Dr. Bruce H. Barnes
and
Frederick R. Stocker
The Pennsylvania State University

INTRODUCTION

One of the most difficult tasks for any classroom instructor is to diagramatically illustrate mathematical and geometric concepts with a blackboard and chalk as the medium of description. Frequently accuracy gives way to hand waving, especially when the concepts involved are three-dimensional in nature or involve a high degree of mathematical accuracy in representation. That a picture is worth a thousand words or handwaves is as true in the classroom as anywhere else. One solution to this problem is to use the computer to supply a textbook base of programs which provide graphical output with a high degree of visual and mathematical accuracy. With static motion* or dynamic motion added, third (structural for example) and fourth (rotation in time for example) dimensional concepts can be made to stand out. This discussion offers some approaches to the use of computer graphics, as an answer to the blackboard and chalk problem. The use of graphics in elementary mathematics is presented by matrices and transformations, while Fourier Series is used as a representative of advanced mathematics.

COMPUTER GRAPHICS

A large diversity of hardware has been developed or used in support of graphic image presentation including cameras and film, overhead projectors, computer high-speed line printers, pen and ink plotters and cathode ray tube displays. Many techniques have been developed to present information for assimilation using devices such as those mentioned. Educational television makes good use of film. Computer assisted instruction uses some combinations of the above mentioned devices. Many college courses have made use of such devices to some extent. This discussion will center around the high speed line printers, pen and ink plotters and cathode ray tube devices as mediums of communication.

Every computer installation has some type of line printer for output. These devices print lines of text on paper varying from standard 8½ by 11" to large format computer printout pages. If the printed lines are patterned or overstruck with several characters, then plots may be generated or geometric shapes approximated. Current devices in use typically output several hundred lines of characters per minute (with overprinting the number of actual lines comes down). Each page of output may be likened to a single frame of film hence enabling a sequence of pages to tell a story as is the case with a sequence of frames on film. The primary advantage to this medium of output is availability and cost. Also considerable software support programming has been done for such devices.

Typical pen and ink plotters allow a pen to be moved over paper and if pressed to the paper a line results from the movement (called a draw), while if the pen is not pressed to the paper no line results on movement, (called a move). Thus, again, plots of geometric shapes may be generated. Likewise, several pages of plotter output may be put together to tell a story as indicated earlier. The principal advantage of such devices is accuracy of detail and large output format. Both the printers and the plotters described take time in the order of seconds to minutes to produce each plot.

The CRT devices have a beam of electrons which may be deflected at variable speed across the phosphor coating on a screen causing the phosphor to glow for a period of time again producing a picture or frame of output. A sequence of such pictures can again be used to tell a story. Once the phosphor is excited, if it continues to emit visable light for a period of minutes the scope employs what is known as a storage tube which holds an image once drawn through use of a high persistance phosphor. If the image fades within a matter of microseconds then the tube utilized is a refresh tube and it employs a low persistance phosphor. Systems with storage tubes allow for viewing single frames in a sequence one at a time with a time delay between frames on the order of seconds to minutes depending on the system employed. Such systems may be used to show static motion as defined earlier. Systems with refresh tubes typically employ more sophisticated hardware to support the rapid updating of information (in the form of pictures and text). Such systems are relatively expensive; however, due to the rapidity with which each picture can be drawn (typically 1/40 second or less) and due to the interaction possible, dynamic motion may be employed and interactive techniques developed to allow the user to communicate with a program in real-time. For a discussion of Computer Graphics see Newman and Sproull (1).

COMPUTER SYSTEM

The line printer on which these pictures were developed is an IBM 1403 printer. The plotter on which these pictures were plotted is a Calcomp 564-30 inch incremental plotter. The interactive graphics system on which these programs were developed is an Adage Model 30 with 32768-30 bit words of core. (2) Control devices used include a joystick, 6 variable control dials, alphanumeric keyboard, 18 function switches and foot-pedals. With this equipment the images are dynamically manipulable by both the instructor and the student.

The programs and techniques discussed in this paper have been tested, implemented and are currently running as part of a computer graphics training program on the Pennsylvania State University Computation Center Adage Model 30. They form part of a computer graphics on-line text book base for the development of computer graphics geometric conceptualization as required by the Computer Science undergraduate course in graphics presented by Penn State.

*Incremental motion where the increments taken show large changes in information or geometric orientation.

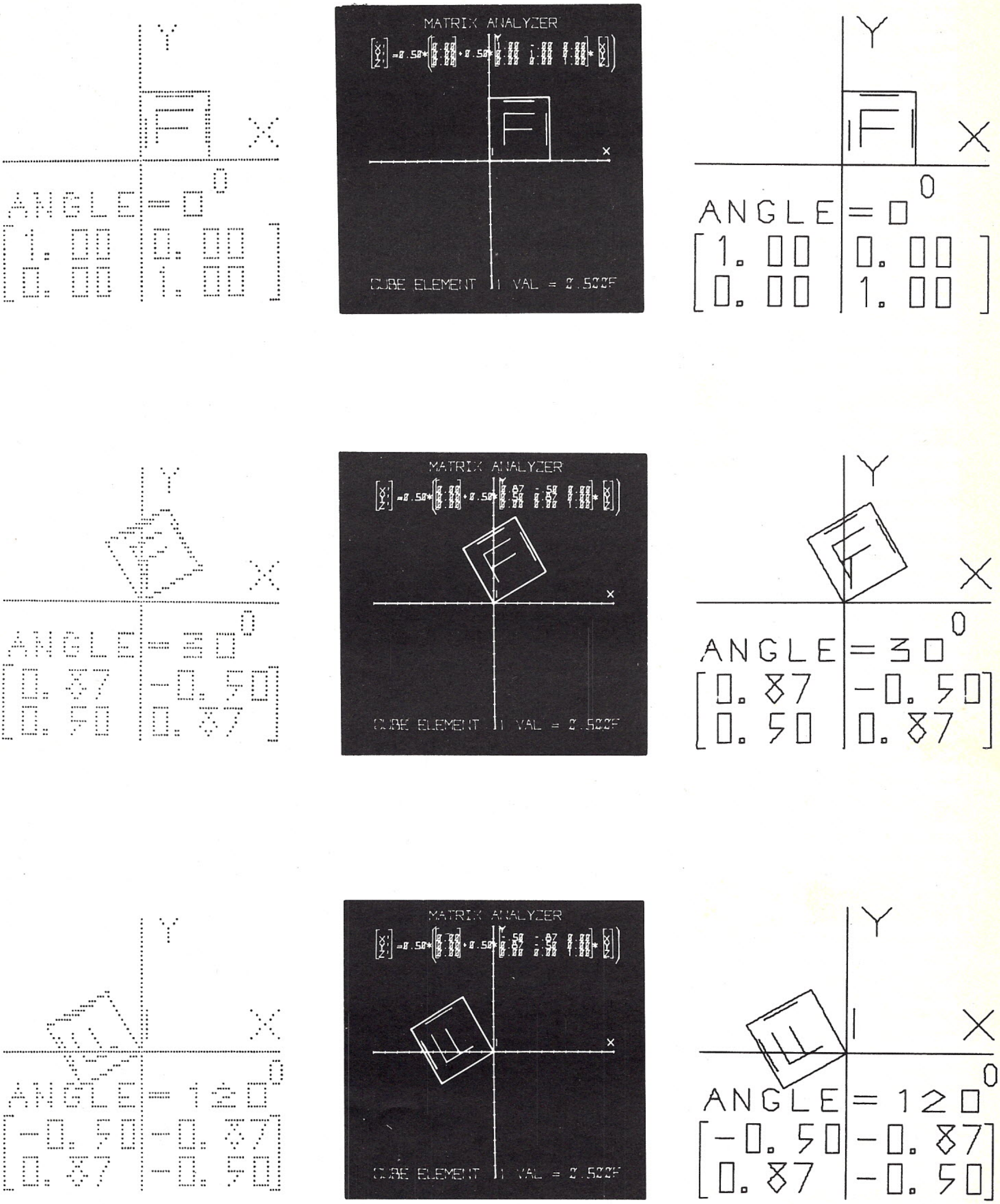

MATRIX ANALYZER
CUBE ELEMENT 1 VAL = 0.500F
Y
X
ANGLE = 0°
1.00 0.00
0.00 1.00
ANGLE = 30°
0.87 -0.50
0.50 0.87
ANGLE = 120°
-0.50 -0.87
0.87 -0.50

Figure 1

They have been successfully used to supplement 9 lectures (one per week) per ten week term (Spring and Fall, 1973 and Spring, 1974). The information was presented directly via the graphics system using the scope as a blackboard and the instructor as the knowledgeable programmer. This course is open to students from most programs in the University.

It should be emphasized that with recent advances in computer hardware, and industry projection of further developments the availability of interactive graphics units is rapidly on the rise. Storage tubes can now be purchased for under $4,000.00. Some mini-computers for programming support are in the one to ten thousand dollar range. Refresh scopes are also becoming inexpensive and raster systems (TV like systems) are being employed for which the costs are within the $10,000 range per unit with mini-computers interfaced for support. These costs will continue to come down as the new miniturization technology, such as used in the new pocket calculators, advances. As this happens, more departmental requests for plotting support in terms of both hardware and software programming will be honored if need can be demonstrated.

PROGRAMS FOR CONCEPT DEMONSTRATION

The computer programs, MATRIX and SQWAV, are utilized in this paper to demonstrate the techniques under discussion. MATRIX demonstrates the geometric significance of matrices and vectors. SQWAV allows sine curve approximations to a square wave to be explored. The MATRIX program is intended to aid in developing understanding at the geometric and analytical levels, thus enabling the instructor or student to develop mathematical comprehension. The SQWAV program illustrates a mechanism whereby detailed analysis for comprehension in an application may be developed. The MATRIX and SQWAV programs provide a variety of visual options to choose from thus freeing the instructor to select the concept area desired within the framework of the program being processed. Through the use of such programs, the comprehension of the student is readily tested and the development of new understanding as well as the correction of weak areas can be accomplished by the instructor. In this manner we feel graphics is seen to be an important teaching aid to the classroom instructor and for the student.

In Figure 1 the interactive scope, plotter and line printer are contrasted at the pictorial level to emphasize realizable aspects of the thoughts presented. The other figures illustrate other concepts which could be viewed in a similar manner.

MATRIX PROGRAM

By selecting a transformation to be varied off a variable control dial or through a program input READ statement, such as rotation about the Z axis (in the XY plane), the variation of vector and matrix equation entries may be monitored dynamically as illustrated in Figure 1. It should be noted that the X,Y-planar rotation submatrix varies but that all Z rotation elements remain unchanged. This allows the two dimensional transformation (rotation) to be observed within the larger context of the three-dimensional environment and also indicates the mathematical independence of the third dimension. In a similar manner, other rotations, translations and scaling may be studied as well as how they effect each other in combination.

EFFECTS OF INDIVIDUAL MATRIX ENTRIES

Another important aspect illustrated in the matrix program is the effect on an individual matrix element on a geometric transformation. Figure 2 presents sequences of pictures which illustrate the changing of the rotational matrix element in row 1, column 1 of the rotation array to affect scaling in X. The scaling range is illustrated from zero to full scale. Figure 3 illustrates the individual and combined effects of skewing in X, then Y and then both X and Y. The last picture of this sequence illustrates viewing such a skewing operation in 3-dimensions. Figure 4 indicates a precision coordinate grid available to check on the transformation accuracy from a visual standpoint, while figure 5 indicates how the various mathematical representations are selectable and comparable on the scope face as different transformational aspects are being monitored. The input and control tables may optionally be viewed as illustrated. Note that in all of the scope illustrations, the primary value being monitored is shown at the bottom of the graphic image to three significant figures while the matrix items being monitored are shown only to two figures, thus illustrating the round off in effect on the system.

SQUARE WAVE PROGRAM

The blackboard problem is very pronounced in teaching Fourier Series. With simple sketching the instructor can illustrate how the terms are summed and possibly convince the student that the series will converge. The accuracy and speed of computer graphics, however, allow the instructor to illustrate more concepts convincingly. On a cathode ray tube one can watch the series converging and gain some meaningful insight into the rapidity of convergence. The Gibb's phenomenon which is almost impossible to sketch with any accuracy or to even convince the students that it exists, is beautifully illustrated (note Figure 6). That is one reason for using the split screen to show Gibb's phenomenon along with the next term to be added and the sum of the first terms of the series. The computer output can be used by each individual student or they can be used as visual aids for the instructor. Both techniques have been used by Mathematics Professors in teaching the Course Fourier Series and Partial Differential Equations to establish understanding which cannot be gained from the usual classroom exercises.

CONCLUSION

Through the transformation matrix and square wave examples we have indicated how programs at both the structure and application levels can be used to produce a symbiotic affect between theory and applications where the student develops understanding in both areas simultaneously. Programs such as this are intended to provide the knowledgeable instructor with a source of mathematical and geometrical information which makes available for selection different representations of the same concepts. A collection of such programs provides an interactive textbook-like data base for teaching fundamental concepts in an area, in this case two and three dimensional transformations contrasting mathematical models with both matrix and geometric representations. Sequences of output as shown in the figures can be obtained from the computer with the student supplying the input sequence for control of an entry in the matrices or an axis or axies about which the rotation is to be observed. The sequential output of pictures will then tell the story. Preferred sequences of values may be selected to develop course material of an instructor's own choosing for classroom work. In the case of an interactive system both the knowledgeable instructor and the novice student may dynamically manipulate the geometric imagery and/or the matrix entries to study the effects of modifying either the geometric structure or the analytical mathematical model which corresponds to the geometric image. In this manner the student develops a visual geometric understanding of the analytic model as well as an analytical understanding of the geometric structures and transformations.

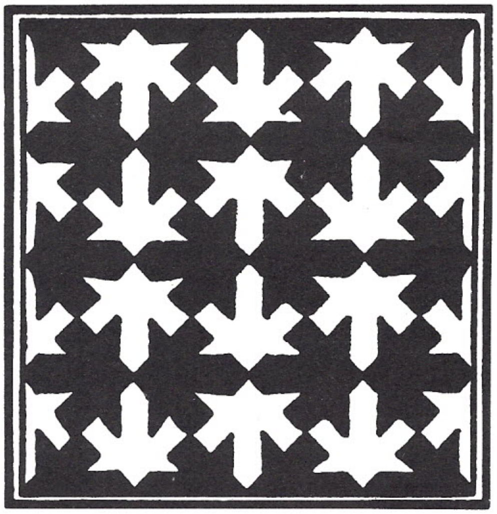
Thus computer graphics can be used as a valuable teaching aid at either the classroom level, the individual (1 to 1) confrontation level, or at the level where the student seeks information through direct program interaction. With a little imagination, such techniques as these can be implemented using any available high speed printers or plotters for output.

It should be noted that programs, such as those illustrated herein, make excellent student programming projects. They allow the use of as much programming skill and imagination as the student is capable of, because any project of this type is open ended and can grow from very simple to quite complex. The end results of a project are useful, giving the student a sense of accomplishment in not doing just another exercise.

REFERENCES

(1) William M. Newman and Robert F. Sproull, *Principles of Interactive Computer Graphics*. McGraw Hill, 1973

(2) *Adage Fortran Programming System*, Adage, Inc., 1079 Commonwealth Avenue, Boston, Mass. 1972



Fluctuation of attention Illusion
Gaze steadily and will change from white design on black background to black design on white background.

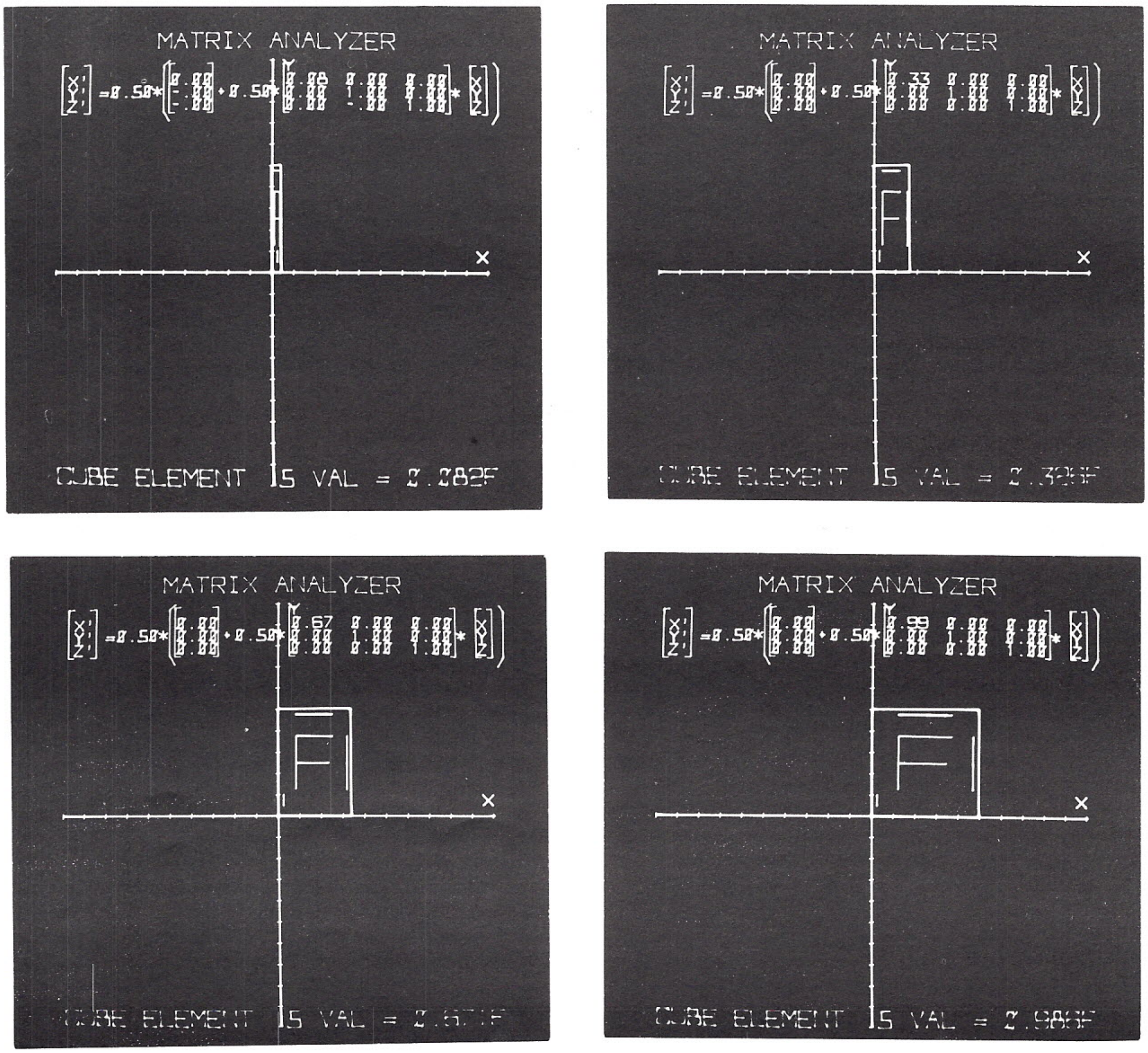


Figure 2

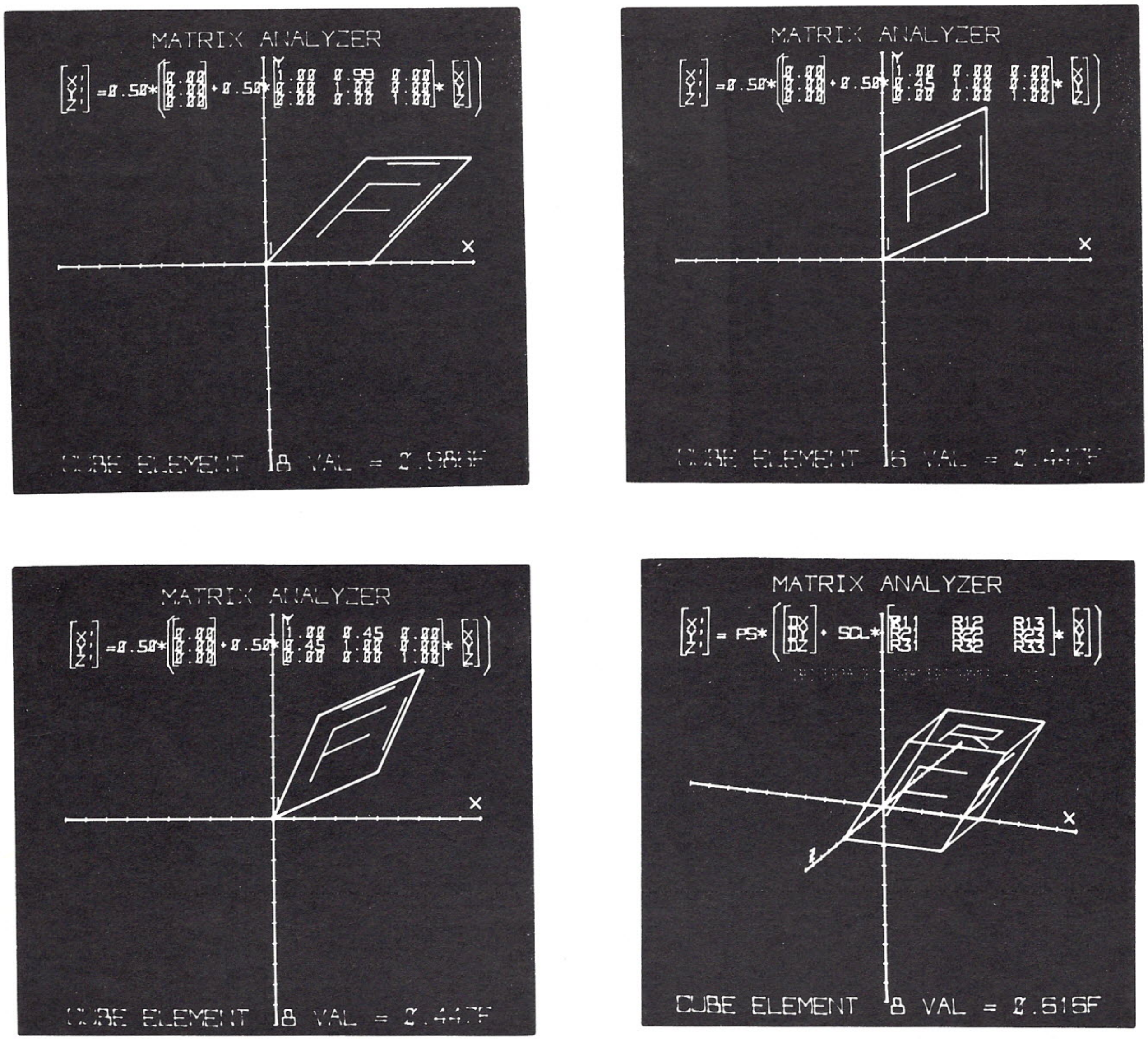

MATRIX ANALYZER
X' Y' Z' =Ø.5Ø*(Ø.ØØ Ø.ØØ Ø.ØØ + Ø.5Ø* 1.ØØ Ø.98 Ø.ØØ Ø.ØØ 1.ØØ Ø.ØØ Ø.ØØ Ø.ØØ 1.ØØ * X Y Z)
X
CUBE ELEMENT 8 VAL = Ø.988F
MATRIX ANALYZER
X' Y' Z' =Ø.5Ø*(Ø.ØØ Ø.ØØ Ø.ØØ + Ø.5Ø* 1.ØØ Ø.ØØ Ø.ØØ Ø.45 1.ØØ Ø.ØØ Ø.ØØ Ø.ØØ 1.ØØ * X Y Z)
X
CUBE ELEMENT 6 VAL = Ø.447F
MATRIX ANALYZER
X' Y' Z' =Ø.5Ø*(Ø.ØØ Ø.ØØ Ø.ØØ + Ø.5Ø* 1.ØØ Ø.45 Ø.ØØ Ø.45 1.ØØ Ø.ØØ Ø.ØØ Ø.ØØ 1.ØØ * X Y Z)
X
CUBE ELEMENT 8 VAL = Ø.447F
MATRIX ANALYZER
X' Y' Z' = PS*(DX DY DZ + SCL* R11 R12 R13 R21 R22 R23 R31 R32 R33 * X Y Z)
X
CUBE ELEMENT 8 VAL = Ø.616F

Figure 3

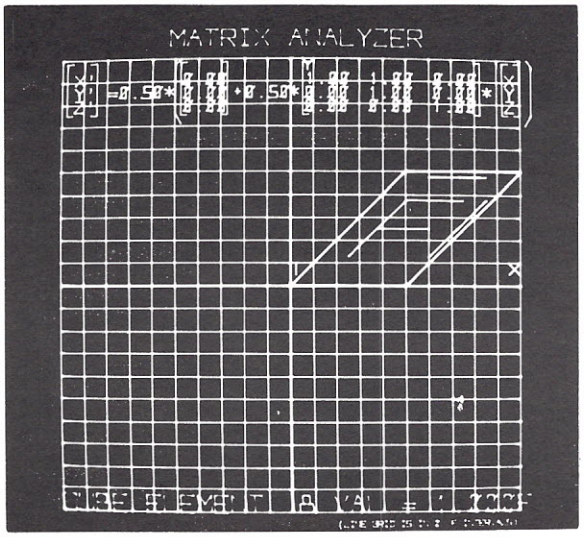

MATRIX ANALYZER
X' Y' Z' =Ø.5Ø*(Ø.ØØ Ø.ØØ Ø.ØØ + Ø.5Ø* 1.ØØ 1.ØØ Ø.ØØ Ø.ØØ 1.ØØ Ø.ØØ Ø.ØØ Ø.ØØ 1.ØØ * X Y Z)
X
CUBE ELEMENT 8 VAL = 1.ØØØF

Figure 4

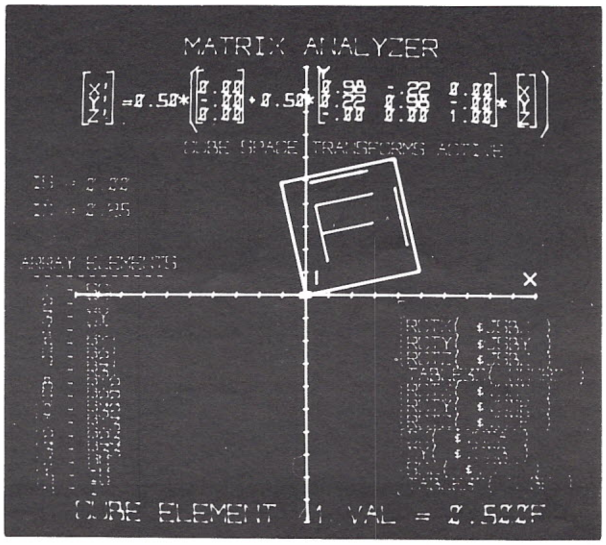

MATRIX ANALYZER
X' Y' Z' =Ø.5Ø*(Ø.ØØ Ø.ØØ Ø.ØØ + Ø.5Ø* Ø.98 -.22 Ø.ØØ Ø.22 Ø.96 Ø.ØØ Ø.ØØ Ø.ØØ 1.ØØ * X Y Z)
CUBE SPACE TRANSFORMS ACTIVE
ARRAY ELEMENTS
X
CUBE ELEMENT 1 VAL = Ø.522F

Figure 5

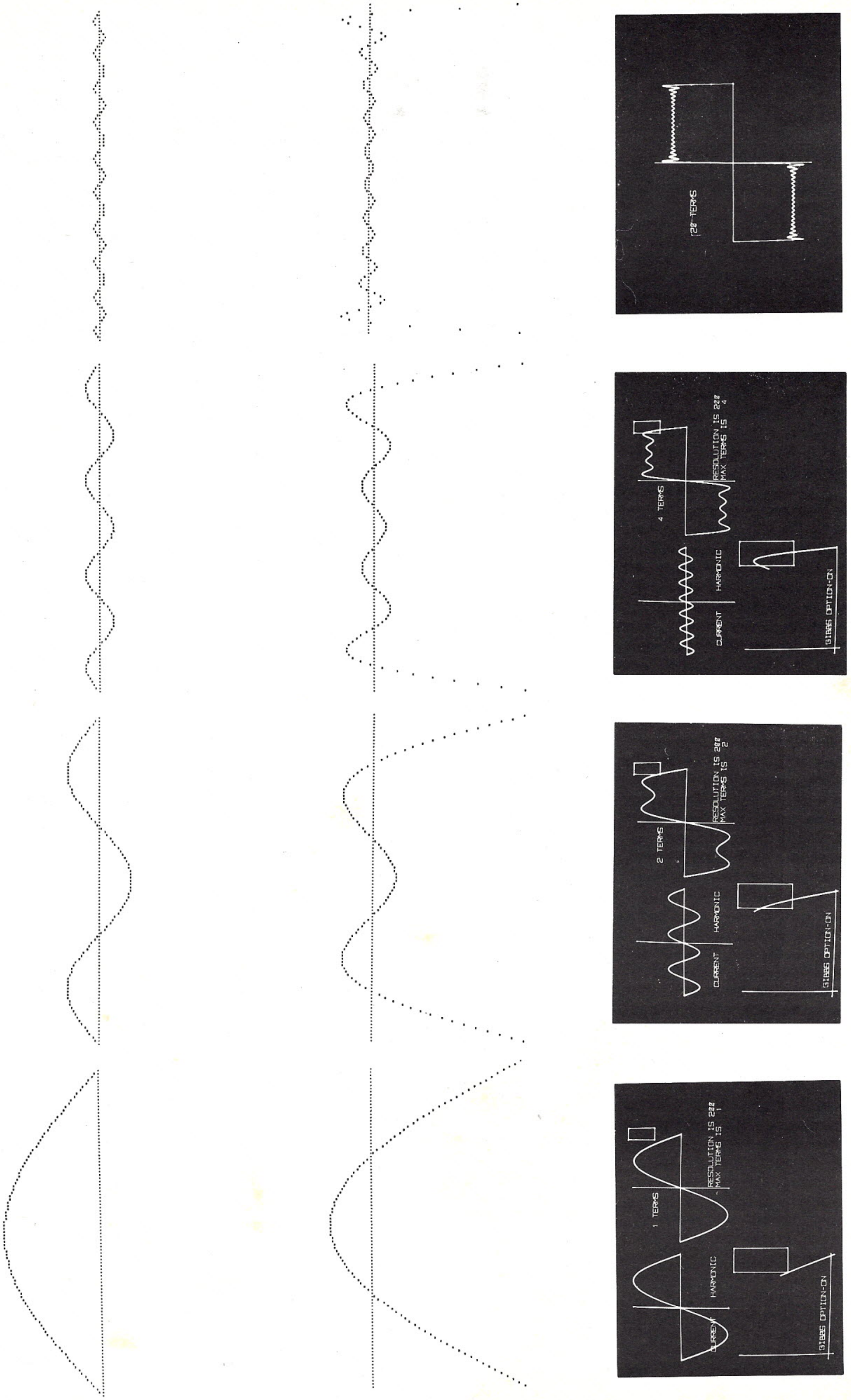

1 TERMS
CURRENT
HARMONIC
RESOLUTION IS 2ØØ
MAX TERMS IS 1
GIBBS OPTION-ON
2 TERMS
CURRENT
HARMONIC
RESOLUTION IS 2ØØ
MAX TERMS IS 2
GIBBS OPTION-ON
4 TERMS
CURRENT
HARMONIC
RESOLUTION IS 2ØØ
MAX TERMS IS 4
GIBBS OPTION-ON
2Ø-TERMS

Figure 6

CMAPS: A Basic Language Program for Choropleth Mapping

James W. Cerny
University of New Hampshire

The best synonym for the cartographer's term "choropleth map" would be "shading-by-area map". This is the familiar type of map in which there is a single data value for each observation area (state, county, census tract, etc.). Typically this involves arraying our data in ascending order so we can classify it into data classes or intervals, with a different shading symbol assigned to each data class. The map is constructed by taking each data value, looking up the appropriate data class, and shading the area with the associated shading symbol for that data class. It is a conservative mapping technique, because we assume the data values apply uniformly throughout each observation area, making no inferences about gradients. A general introduction to this and other types of mapping can be found in Robinson and Sale (1969).

M. W. Scripter (1969) devised a Fortran computer program for choropleth mapping on computers with limited storage. Since then, many Fortran versions of his program have been produced (Caspall and Jozwiak, 1973; Whitmore, 1972). Listed and discussed below, however, is CMAPS, a Basic-language version of Scripter's program designed for interactive computing on a terminal. The program, as presented, is only of moderate length and complexity and many embellishments are possible.

Fundamental to computer mapping is some method of coding information to describe the base area to be mapped. CMAPS uses a scan-line technique as follows. First, you must select a base map of a suitable scale. The areas on this base map will be approximated using the print symbols available on the terminal. Since most terminals print (or display) 10 characters per inch horizontally and 6 characters per inch vertically, you should overlay the base map with a 6 x 10 to the inch grid. Each row in this grid will be one row in the printed map. Obviously very small areas will be difficult to represent without a map of sufficiently large scale. Each row will be composed of segments, one segment representing a slice of a map area or the background surrounding the map. Map areas should be numbered in an arbitrary but consistent manner, either from top to bottom or from an alphabetical list of the areas. The background is assigned the number N+1, where N is the number of map areas. Each segment requires two pieces of information to describe it to the CMAPS program: the number of the map area it represents and the rightmost grid cell (print position) it occupies in that scan-line row. A whole scan-line data file is thus constructed for each base area that will be mapped. Each scan-line in that file consists of the number of segments in that line and then, in pairs, the segment number and rightmost print position. The leftmost print positions are deduced by CMAPS. You should also realize that to change the size of the printed map requires construction of a new scan-file.

The version of CMAPS listed here (Figure 1) allows an unlimited number of scan-lines or length of the map, up to 20 map areas (not counting background), up to 20 data classes, and will produce a map up to 72 characters wide. Before running CMAPS a scan-line file as described above must have been created. Then the program asks for a map title, map parameters, data class limits, print symbols for the data classes, and the data itself. In the sample run shown (Figure 2), a scan-line file of New Hampshire was used that was 50 scan-lines long, 45 columns wide, and composed of 10 map areas or counties (Figure 3). The data mapped were the percentage of the 1970 population in each county that was either first or second generation Canadian. The county order for numbering in the scan deck, from 1 through 10, was Coos, Grafton, Carroll, Belknap, Sullivan, Merrimack, Strafford, Rockingham, Cheshire, and Hillsborough. The program operates as follows. It classifies the data according to the data values and class limits read in. If a value is on the border between two classes, it is assigned to the lower data class. Then CMAPS begins a loop, reading information from the scan file one line at a time. It never stores more than one line of the map at a time. It reads how many segments there are for that line, and for each segment reads which map area that is, deduces which contiguous print positions belong to that segment, computes which map symbol should be associated with the segment on the basis of the data classification already done, proceeding until it has composed a whole line, which is then printed.

In the sample run (Figure 2) the highest values are in the northernmost county (Coos) and in the counties where there was the greatest industrialization in the nineteenth century. Most of the people of Canadian origin in New Hampshire are of French background, drawn to the prospect of work in the mills and factories when a different economic situation existed. Note that an attempt has been made to use darker shading for higher values or percentages. A different map classification might show more or less detail to the pattern.

In our implementation of CMAPS, we use a separate program to print an explanatory description of CMAPS and choropleth mapping, but such information could be incorporated directly in the CMAPS program. We also have a little utility program called SCANMAKE (Figure 4) to facilitate the construction of scan files by people who are novices. CMAPS could be made more retentive, storing shading and data values so that users would only have to supply new data class limits to experiment in the mapping of a set of data. At the risk of greatly enlarging memory requirements, the scan-file information could be read in and stored in an array, or it could be stored in data statements. In making modifications perhaps two things should be kept in mind. People are not likely to seek finished quality maps from CMAPS, but it can be very useful for working maps and quick proofing of data classifications. It is time consuming to print large numbers of maps on the same base at a terminal and perhaps that type of production work is better done in a batch environment.

References

Caspall, F. C., and Jozwiak, L. 1973. "CORMAP: A Choroplethic Mapping Routine," *Technical Report No. 1,* Institute for Regional, Rural, and Community Studies, Western Illinois University.

Robinson, A., and Sale, R. 1969. *Elements of Cartography,* 3rd ed. New York: Wiley & Sons.

Scripter, M. W. 1969. "Choropleth Maps on Small Digital Computers," *Proceedings,* Association of American Geographers, v. 1, 133–136.

Whitmore, S. E. 1972. "INMAP: A Computer-Aided Instruction Program for Teaching Class-Interval Selection," pp. 39–42 in *Proceedings,* Conference on Computer Applications in Geography, 12–14 October 1972, Department of Geography, SUNY at Albany.

```
100 REM CMAPS . . . CHOROPLETH MAPPING PROGRAM.
110 REM JAMES W. CERNY, MARCH 1975.
120 REM REFERENCES: (1) SCRIPTER, M.W.1969. "CHOROPLETH MAPS ON
130 REM SMALL DIGITAL COMPUTERS," PROCEEDINGS, ASSOCIATION OF
140 REM AMERICAN GEOGRAPHERS, V. 1, 133-136.  (2) CASPALL, F.C. AND
150 REM JOZWIAK, L. 1973. "CORMAP: A CHOROPLETHIC MAPPING ROUTINE,"
160 REM TECHNICAL REPORT NO. 1, INSTITUTE FOR REGIONAL, RURAL, AND
170 REM COMMUNITY STUDIES, WESTERN ILLINOIS UNIVERSITY.
180 DIM A(72),B(72),C$(22),D(20),E$(72),U(21),X(20)
190 PRINT "PROGRAM CMAPS MAKES CHOROPLETH MAPS.  SEE **CMAPSDOC."
200 PRINT
210 PRINT "ENTER THE MAP TITLE (UP TO 18 CHARS)."
220 INPUT F$
230 PRINT "ENTER THE NO. OF CLASSES, AREAS, SCAN LINES,"
240 PRINT "AND MAP WIDTH IN COLUMNS."
250 INPUT N1,N2,N3,L
260 N4=N1+1
270 PRINT "ENTER THE LOWER LIMIT FOR THE FIRST CLASS AND,"
280 PRINT "IN ORDER, THE UPPER LIMIT FOR EACH CLASS."
290 MAT INPUT U(N4)
300 U(N4)=U(N4)*.00001+U(N4)
310 REM ASSIGN BACKGROUND AND OUT-OF-RANGE SYMBOLS.
320 C$(N1+1)='.'
330 C$(N1+2)='R'
340 PRINT "ENTER A PRINT SYMBOL FOR EACH CLASS, ONE AT A TIME,"
350 PRINT "IN ORDER.  THE SYMBOLS '.' AND 'R' ARE RESERVED."
360 FOR M=1 TO N1
370 INPUT C$(M)
380 NEXT M
390 PRINT "ENTER THE DATA TO BE MAPPED, ONE VALUE FOR EACH"
400 PRINT "MAP AREA.  THE ORDER MUST MATCH SCAN DECK NUMBERING."
410 MAT INPUT X(N2)
420 PRINT "WHAT IS THE NAME OF YOUR SCAN-LINE FILE?"
430 INPUT G$
440 OPEN 1,G$,INPUT
450 REM CLASSIFY MAP AREAS ACCORDING TO DATA VALUES.
460 D(N2+1)=N4
470 FOR M=1 TO N2
480 IF X(M)< U(1) THEN 530
490 FOR N=2 TO N4
500 Q=N-1
510 IF U(N) > X(M) THEN 540
520 NEXT N
530 Q=N1+2
540 D(M)=Q
550 NEXT M
560 REM BEGIN SEQUENCE TO COMPOSE AND PRINT MAP LINES.
570 FOR N=1 TO 5
580 PRINT
590 NEXT N
600 PRINT F$
610 PRINT
620 FOR M=1 TO N1
630 PRINT "CLASS";M;"SYMBOL ";C$(M);"   LIMITS";U(M);"TO";U(M+1)
640 NEXT M
650 PRINT
660 PRINT
670 FOR M=1 TO N3
680 REM FETCH NO. SEGMENTS PER LINE, THEN BY PAIRS,
690 REM AREA NO. AND RIGHMOST PRINT POSITION FOR EACH.
700 GET 1: P1
710 FOR N=1 TO P1
720 GET 1: A(N),B(N)
730 NEXT N
740 I1=0
750 FOR N=1 TO P1
760 I2=I1+1
770 I1=B(N)
780 K=A(N)
790 K=D(K)
800 FOR J=I2 TO I1
810 E$(J)=C$(K)
820 NEXT J
830 NEXT N
840 IF I1>= L THEN 890
850 I2=I1+1
860 FOR J=I2 TO L
870 E$(J)=C$(N4)
880 NEXT J
890 L1=L-1
900 FOR N=1 TO L1
910 PRINT E$(N);
920 NEXT N
930 PRINT E$(L)
940 NEXT M
950 REM END COMPOSE AND PRINT SEQUENCE.
960 CLOSE 1
970 FOR N=1 TO 5
980 PRINT
990 NEXT N
1000 PRINT "DO YOU WANT TO MAKE ANOTHER MAP (YES OR NO)?"
1010 INPUT A$
1020 IF A$='YES' THEN 210
1030 STOP
1040 END
```

Figure 1. Program CMAPS Listing.

```
PROGRAM CMAPS MAKES CHOROPLETH MAPS.  SEE **CMAPSDOC.

ENTER THE MAP TITLE (UP TO 18 CHARS).
?PCT 1&2 CANADIAN
ENTER THE NO. OF CLASSES, AREAS, SCAN LINES,
AND MAP WIDTH IN COLUMNS.
?3,10,50,45
ENTER THE LOWER LIMIT FOR THE FIRST CLASS AND,
IN ORDER, THE UPPER LIMIT FOR EACH CLASS.
?0,10,20,35


ENTER A PRINT SYMBOL FOR EACH CLASS, ONE AT A TIME,
IN ORDER.  THE SYMBOLS '.' AND 'R' ARE RESERVED.
?0
?X
?#
ENTER THE DATA TO BE MAPPED, ONE VALUE FOR EACH
MAP AREA.  THE ORDER MUST MATCH SCAN DECK NUMBERING.
?30.7,9.6,7.3,12.8,12.0,10.5,12.4,8.7,7.7,17.1


WHAT IS THE NAME OF YOUR SCAN-LINE FILE?
?NHSCAN
```

Figure 2. Sample CMAPS Run.

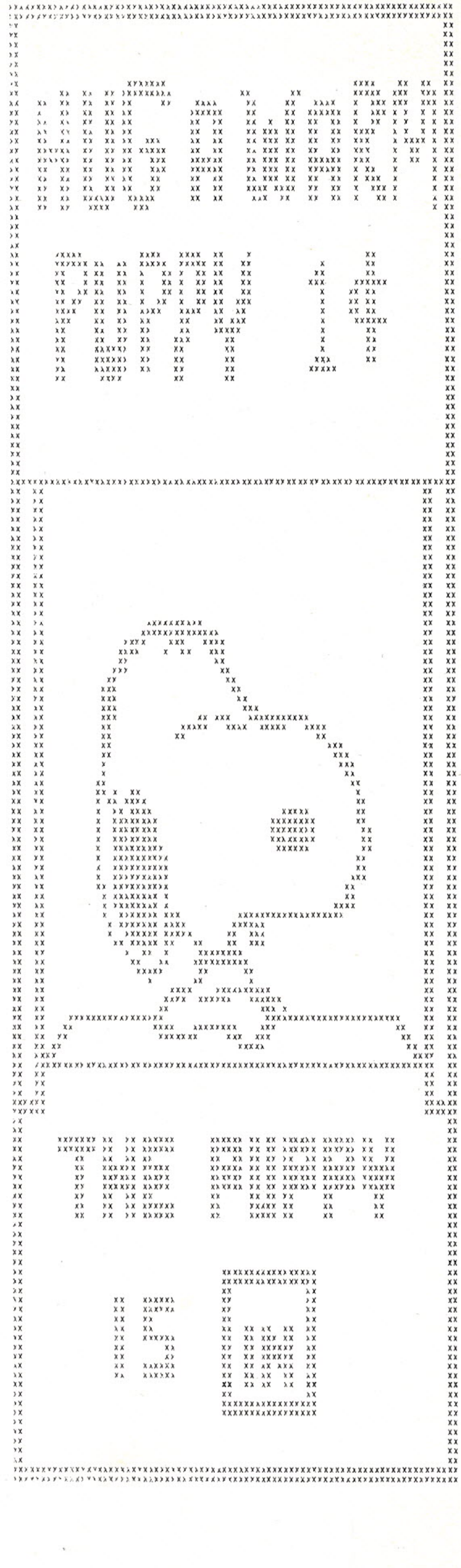
```
PCT 1&2 CANADIAN

CLASS 1     SYMBOL 0    LIMITS 0     TO 10
CLASS 2     SYMBOL X    LIMITS 10    TO 20
CLASS 3     SYMBOL #    LIMITS 20    TO 35.0003


.............................................
.............................................
..............................##...#.........
............................########.........
............................########.........
...........................#########.........
...........................#########.........
..........................##########.........
..........................##########.........
..........................###########........
.........................############........
........................#############........
........................#############........
.........................############........
..........................###########........
.........................############........
........................#############........
.......................##############........
....................00###############........
.................000000##############........
...............0000000000############0.......
...............00000000000000###000000.......
...............000000000000000#0000000.......
...............00000000000000000000000.......
..............000000000000000000000000.......
..............000000000000000000000000.......
.............0000000000000000000000000.......
............00000000000000000000000000.......
............00000000000000000000000000.......
...........000000000000000000000000000.......
.........00000000000000000XX0000000000.......
.........000000000000000XXXXXX00000000.......
........000000000000000XXXXXXX00000000.......
.......XXXXXX00000XXXXXXXXXXXXXXXXX0000......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX....
......XXXXXXXXXXXXXXXXXXXXXXXX000XXXXXXXXX...
......X00000XXXXXXXXXXXXXXXXXXX00000XXXXXX...
......000000000XXXXXXXXXXXXXXX000000XXXXX00..
......000000000XXXXXXXXXXXXXX000000000000000.
.....0000000000XXXXXXXXXXXXXX000000000000000.
....00000000000XXXXXXXXXXXXX000000000000000..
...0000000000000XXXXXXXXXXXX000000000000.....
....000000000000XXXXXXXXXXXXX000000..........
.....000000000000XXXXXXXXXXXXXX000...........
......000000000000XXXXXXXXXXXXXX.............
.............................................




DO YOU WANT TO MAKE ANOTHER MAP (YES OR NO)?
?NO
```

Figure 3. NHSCAN File.

```
LINE NO. 1          1     11    45
LINE NO. 2          1     11    45
LINE NO. 3          5     11    30    1     32    11    35    1     36
 11    45
LINE NO. 4          3     11    28    1     36    11    45
LINE NO. 5          3     11    28    1     36    11    45
LINE NO. 6          3     11    27    1     36    11    45
LINE NO. 7          3     11    27    1     36    11    45
LINE NO. 8          3     11    26    1     36    11    45
LINE NO. 9          3     11    26    1     36    11    45
LINE NO. 10         3     11    26    1     37    11    45
LINE NO. 11         3     11    25    1     37    11    45
LINE NO. 12         3     11    24    1     37    11    45
LINE NO. 13         3     11    24    1     37    11    45
LINE NO. 14         3     11    25    1     37    11    45
LINE NO. 15         3     11    26    1     37    11    45
LINE NO. 16         3     11    25    1     37    11    45
LINE NO. 17         3     11    24    1     37    11    45
LINE NO. 18         3     11    23    1     37    11    45
LINE NO. 19         4     11    20    2     22    1     37    11    45
LINE NO. 20         4     11    17    2     23    1     37    11    45
LINE NO. 21         5     11    15    2     25    1     37    3     38
 11    45
LINE NO. 22         5     11    15    2     28    1     32    3     38
 11    45
LINE NO. 23         6     11    15    2     29    3     30    1     31
 3     38    11    45
LINE NO. 24         4     11    15    2     29    3     38    11    45
LINE NO. 25         4     11    14    2     30    3     38    11    45
LINE NO. 26         4     11    14    2     30    3     38    11    45
LINE NO. 27         4     11    13    2     30    3     38    11    45
LINE NO. 28         4     11    12    2     26    3     38    11    45
LINE NO. 29         4     11    12    2     26    3     38    11    45
LINE NO. 30         4     11    11    2     26    3     38    11    45
LINE NO. 31         5     11    9     2     26    4     28    3     38
 11    45
LINE NO. 32         5     11    9     2     24    4     30    3     38
 11    45
LINE NO. 33         5     11    8     2     23    4     31    3     38
 11    45
LINE NO. 34         8     11    7     5     13    2     18    6     22
 4     34    7     35    3     39    11    45
LINE NO. 35         6     11    7     5     17    6     23    4     33
 7     39    11    45
LINE NO. 36         6     11    7     5     15    6     27    4     34
 7     39    11    45
LINE NO. 37         6     11    7     5     14    6     29    4     34
 7     39    11    45
LINE NO. 38         6     11    7     5     14    6     31    4     33
 7     40    11    45
LINE NO. 39         5     11    6     5     14    6     33    7     41
 11    45
LINE NO. 40         6     11    6     5     15    6     31    8     34
 7     42    11    45
LINE NO. 41         9     11    6     5     7     9     12    5     15
 10    18    6     31    8     36    7     42    11    45
LINE NO. 42         8     11    6     9     15    10    23    6     30
 8     36    7     41    8     43    11    45
LINE NO. 43         6     11    6     9     15    10    27    6     29
 8     44    11    45
LINE NO. 44         5     11    5     9     15    10    29    8     44
 11    45
LINE NO. 45         5     11    4     9     15    10    28    8     43
 11    45
LINE NO. 46         5     11    3     9     16    10    28    8     40
 11    45
LINE NO. 47         5     11    4     9     16    10    29    8     35
 11    45
LINE NO. 48         5     11    5     9     17    10    31    8     34
 11    45
LINE NO. 49         4     11    6     9     18    10    32    11    45
LINE NO. 50         1     11    45
```

```
100 REM PROGRAM TO PREPARE SCAN-LINE FILES.
110 REM JAMES V. CERNY, APRIL 1975.
120 PRINT "PROGRAM TO PREPARE SCAN-LINE FILES.  SEE **CMAPSDOC."
130 PRINT
140 DIM B(50)
150 PRINT "FILE NAME?"
160 INPUT A$
170 OPEN 1,A$,OUTPUT
180 PRINT
190 K=0
200 PRINT "NEW ROW.  ENTER NO. SEGS.  ZERO TO STOP."
210 INPUT A
220 IF A=0 THEN 300
230 K=K+1
240 PUT 1: A
250 M=A*2
260 PRINT "ENTER SEGMENT PAIRS: AREA NO., RIGHT PRINT POS."
270 MAT INPUT B(M)
280 MAT PUT 1: B
290 GO TO 200
300 CLOSE 1
310 OPEN 1,A$,INPUT
320 FOR N=1 TO K
330 GET 1: A
340 PRINT A;
350 M=A*2
360 MAT GET 1: B(M)
370 MAT PRINT B
380 NEXT N
390 STOP
400 END
```

Figure 4. Program SCANMAKE.



Creative Programming Techniques. . . .

Heapsort

Most programming texts present the problem of writing one or two basic types of sort programs. Are these generally used in production? Usually not. One of the most efficient production sort algorithms is known as *Heapsort.* In the richly commented BASIC program below, Geoffrey Chase, OSB, of the Portsmouth Abbey School has written a *Heapsort* routine for both character string or numeric sorting. Look it over. Study how it works. And when you want a really efficient sort routine, use it!

```
NOTES:

(1) EVIDENTLY THIS CAN BE SPLIT INTO TWO PROGRAMS; OR YOU
    CAN CUT OUT THE UNNEEDED HALF.

(2) LINE 120 CAN BE DIMENSIONED AS DESIRED.

(3) THE  !  "TAG" COMMENTS AREN'T NEEDED.  SOME BASICS
    ALLOW  !  , SOME ALLOW  '  INSTEAD, SOME NEITHER.
```

```
100 REM.  KNUTH/WILLIAMS/FLOYD  HEAPSORT ALGORITHM.
110 !    PAS '74
120 DIM N(150),C$(150)
130 PRINT
135 PRINT
140 PRINT
145 PRINT "TYPE  C  FOR CHARACTER STRING SORT,"
150 PRINT "TYPE  N  FOR NUMBER SORT. ";
155 INPUT W$
160 N=0                          ! START COUNT=N AT 0
163 PRINT
166 PRINT
170 IF W$="N" THEN 480
180 IF W$<>"C" THEN 140          ! BAD REPLY
190 !--------------------------< CHARACTER SORT: >----
200 GOSUB 720                    ! ASK FOR STOP CODE
210 INPUT S$                     ! GET STOP CODE
215 PRINT
220                              ! INPUT LOOP:
230 N=N+1
235 INPUT C$(N)
240 IF C$(N)<>S$ THEN 230
250                              ! END OF INPUT...
260 N=N-1
265 PRINT
270                              ! HEAPSORT PROPER:
280 L=INT(N/2)+1
285 N1=N                         ! PRESERVE N, USE N1
290 IF L=1 THEN 310
300 L=L-1
303 A$=C$(L)
306 GOTO 350
310 A$=C$(N1)
315 C$(N1)=C$(1)                 ! MOVE TOP OF HEAP TO END
320 N1=N1-1                      ! HEAP IS 1 SMALLER NOW
330 IF N1=1 THEN 440             ! ONLY ONE LEFT? THEN WE!RE DONE.
340                              ! NO, CONTINUE
350 J=L
360 I=J
365 J=2*J                        ! LOOK FOR "SONS" OF I
370 IF J=N1 THEN 400
380 IF J>N1 THEN 420             ! "N1" IS SIZE OF ACTIVE LIST
390 IF C$(J)>=C$(J+1) THEN 400   ! CHOOSE LARGER "SON"
395 J=J+1
400 IF A$>=C$(J) THEN 420
410 C$(I)=C$(J)
415 GOTO 360                     ! LARGER SON REPLACES PARENT
420 C$(I)=A$
425 GOTO 290
430                              ! END OF SORT...
440 C$(1)=A$
450 FOR I=1 TO N
453 PRINT C$(I)                  ! OR REVERSE ORDER: I=N TO 1 STEP -1
456 NEXT I
460 GOTO 130
470 !--------------------------< NUMERIC SORT: >-----
480 GOSUB 720
483 INPUT S
486 PRINT
490 N=N+1
493 INPUT N(N)
496 IF N(N)<>S THEN 490
500 N=N-1
505 PRINT
510     !
520 L=INT(N/2)+1
525 N1=N
530 IF L=1 THEN 550
540 L=L-1
543 A=N(L)
546 GOTO 590
550 A=N(N1)
555 N(N1)=N(1)
560 N1=N1-1
570 IF N1=1 THEN 680
580     !
590 J=L
600 I=J
605 J=2*J
610 IF J=N1 THEN 640
620 IF J>N1 THEN 660
630 IF N(J)<N(J+1) THEN J=J+1   ! FANCY "IF" SYNTAX. COMPARE
640 IF A>N(J) THEN 660                                 390-400.
650 N(I)=N(J)
655 GOTO 600
660 N(I)=A
665 GOTO 530
670     !
680 N(1)=A
690 FOR I=1 TO N
693 PRINT N(I)
696 NEXT I
700 GOTO 130
710 !--------------------------< SUBROUTINE: >-------
720 PRINT "PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR"
730 PRINT "LIST, WHICH WILL ACT AS AN 'END-OF-LIST' SIGNAL: ";
740 RETURN
750     !
760 END
```

```
TYPE  C  FOR CHARACTER STRING SORT,
TYPE  N  FOR NUMBER SORT. ? C


PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR
LIST, WHICH WILL ACT AS AN 'END-OF-LIST' SIGNAL: ? KNUTH

? DAVID AHL, ESQ.
? COSMO COMPUTERS
? ABPLANALP LTD.
? PETRODOLLARS
? DMA TRANSFER
? CREATIVE COMP.
? M.O.S. ABACUS
? ALGORITHMS
? LEONARDO P.
? CHINESE REMS.
? SORTED STRINGS
? NEG. FULLBACK
? STAR TREK, V.2
? KNUTH

ABPLANALP LTD.
ALGORITHMS
CHINESE REMS.
COSMO COMPUTERS
CREATIVE COMP.
DAVID AHL
DMA TRANSFER
LEONARDO P.
M.O.S. ABACUS
NEG. FULLBACK
PETRODOLLARS
SORTED STRINGS
STAR TREK



TYPE  C  FOR CHARACTER STRING SORT,
TYPE  N  FOR NUMBER SORT. ? N


PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR
LIST, WHICH WILL ACT AS AN 'END-OF-LIST' SIGNAL: ? -1E6

? 3.1416
? 22222
? 2E10
? 2E-10
? 66.666
? -1E5
? -1E6

-100000
 2.00000E-10
 3.1416
 66.666
 22222
 2.00000E+10



TYPE  C  FOR CHARACTER STRING SORT,
TYPE  N  FOR NUMBER SORT. ?
STOP AT LINE 155
READY
```

A Comparison of Sorts

John P. Grillo
Dept. of Computer Information Systems
West Texas State University
Canyon, Texas 79016

When students in programming courses compare notes, they find that there is a fairly small set of computer problems which are given as programming assignments in practically all courses. Here is a sampling of these Golden Oldies. *The Indian Problem:* If the Indians had deposited the $24 they got for Manhattan Island in 1620 and earned 6% interest compounded yearly, what would that deposit be worth now? *Fibonacci Numbers:* What is the largest Fibonacci number less than a given number? *Grain of Wheat, or Doubled Penny:* Starting with one grain of wheat, or one penny, and doubling the number every day, how many whatevers are there on the 30th (64th) day? *Table Printing:* Produce as output a table of the [squares, square roots, trig functions] of numbers between 1 and 50. *The Sort:* Sort the data provided in ascending (descending) order and print it.

All of these have innumerable variants and all, except for the sort, are based on relatively obvious, simple, or already familiar looping algorithms which show off the computer's ability to handle simple loops. The sort is a different type of problem: (1) the output can be achieved using any of a number of algorithms, but not any are truly easy to understand; (2) the object of the problem is not to produce the output as much as to learn the algorithm and to optimize computer efficiency by minimizing core use and execution time; (3) the algorithm used is often called a production algorithm, one which is used widely in application programming.

All too often students are presented with the simplest algorithm because it is easiest to learn. That is true enough, but unfortunately, that algorithm is the one students tend to use any time they have to sort, just because they know it. This "horseblinders" result would be of no consequence were the algorithm learned the best one, or even one of the better ones. But the algorithm taught, and learned, is usually the worst one, the bubble sort.

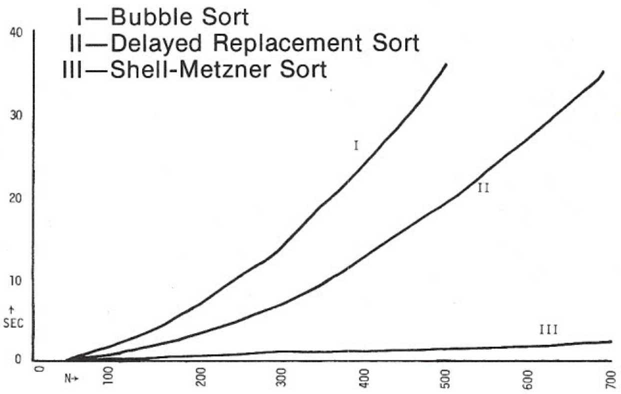
This technique is called bubble sorting because of the way it "floats" the smaller numbers to the top, just like bubbles in a column of water. It might be better called the "trouble sort," though, because of all the machinations that go on at the lower level just to float that number up there.

Slightly better, in terms of efficiency, is the delayed replacement sorting technique. This is really a modification of the bubble sort, except that the smallest of two numbers is not "floated" until it is found to be the smallest of all; whereas the bubble sort floats the smaller of a pair, then checks another pair, the delayed replacement sort checks all pairs and floats only the one found to be the smallest. This greatly reduces the number of executions of the switching statements. The number of pairwise comparisons is exactly equal both in the delayed replacement sort and in the bubble sort, and that number rises exponentially as the number of elements to be sorted rises.

An adaptation by Marlene Metzner (2) of the Shell sort overcomes both difficulties: the number of comparisons is roughly ten times the number of elements to be sorted, and the number of switches is roughly five times the number of elements, if that number of elements is less than 1000. This ratio of comparisons to switches makes intuitive sense, since one would expect that a pair of numbers chosen for possible switching would require switching only half the time.

Appended to this article is a listing of a BASIC program which was used to test sorting algorithms. As an added benefit, the random numbers produced are made to approximate a normal distribution and are truncated. Thus the output from this program can be used as a sample of scores with known statistics. By timing the three methods of sorting using various sample sizes, some estimates of sorting time were calculated. Figure 1 shows graphically the effect of algorithm selection on sorting time.

Figure 1—Observed Sort Times



Figures 2, 3, and 4 are the flowcharts for the three sorting algorithms. All are written to sort a table of N data entries in a table D without use of any additional array space; that is, they are all replacement sorts—array D starts unsorted and becomes sorted. The bubble sort, or Sort I, has as a characteristic feature the use of only two indices, I and J, and no checking of indices except against N, the number of elements being sorted. The delayed replacement sort, Sort II, uses three indices, I, J, and K, and only one of them is compared to N. Note also that in Sort II discovering that D(J) is greater than D(I) does not force a switch; much more index checking is performed first. The Shell-Metzner sort, Sort III, at first glance seems to have regressed to Sort I in that if D(I) is greater than D(L), they are switched. But though this is true, the comparison is performed only after much checking, using not 2, not 3, but 5 indices—I, J, K, L, and M. As a hint in beginning to understand Sort III, consider that the first

Figure 2

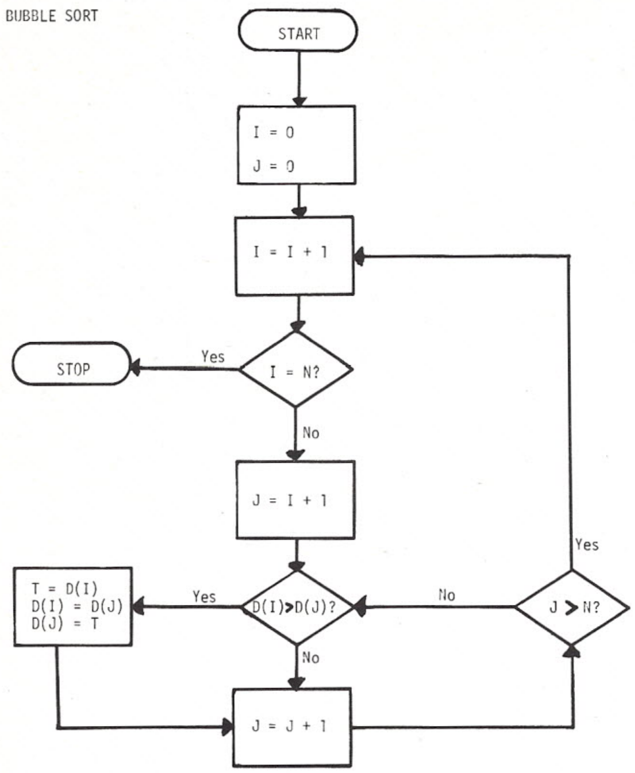


Figure 3

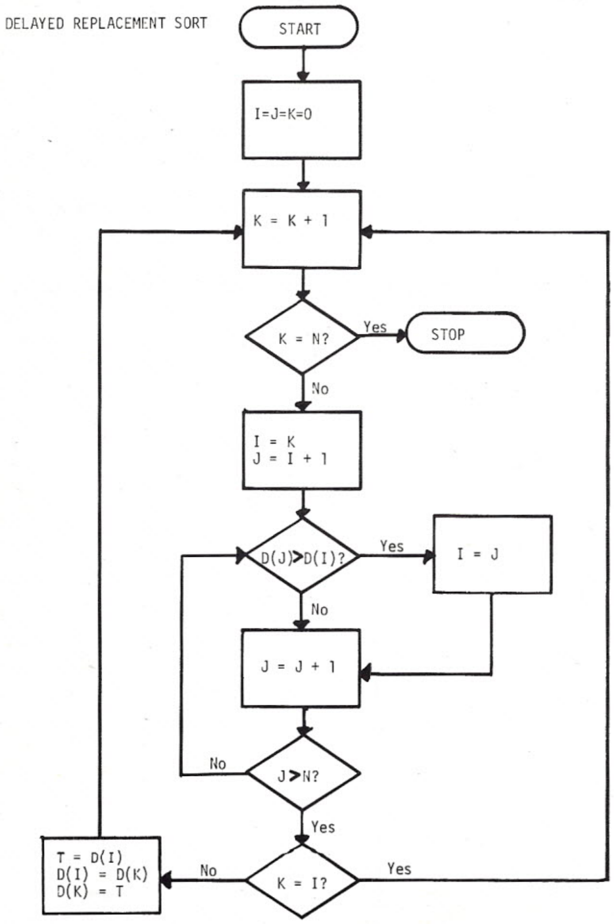
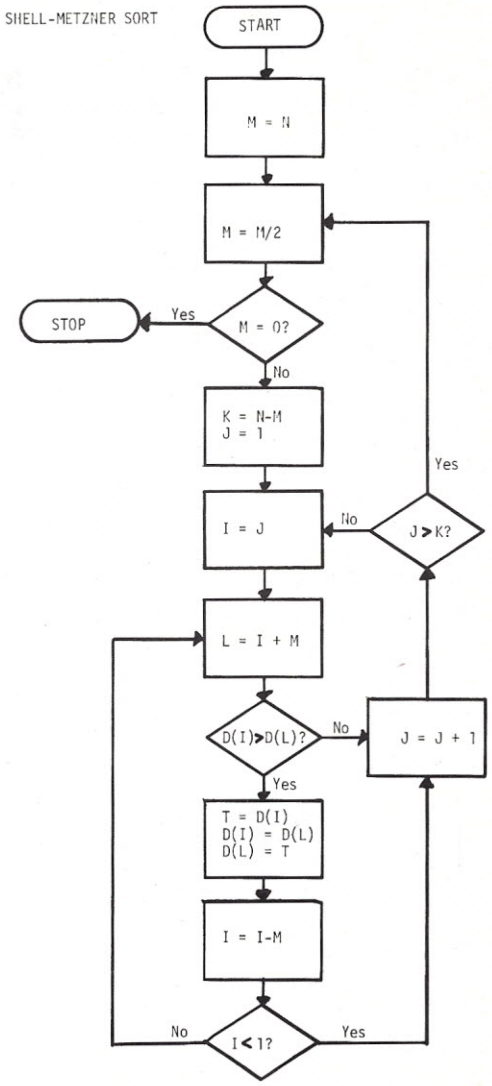


Figure 4



compared pair of 100 numbers is the 1st and 51st; the second is the 2nd and 52nd; etc.

Table I summarizes information on execution of each of the sorts on sets of 10 to 2,000 elements. These elements, or numbers to be sorted, were generated by the program listed in the appendix, so they were normally distributed. The table lists three numbers for each set of elements and each algorithm: T = time of execution in milliseconds on a DECSystem 10 KI processor; S = number of times pairs of elements were switched; C = number of

Table I—Sort Execution Data

		N							Approximate Proportionality to N
		10	20	50	100	200	500	1,000	
SORT I	T	33	84	450	1,700	7,500	34,000	150,000	$.385\ N^{1.84}$
	S	19	100	620	2,700	11,000	63,000	250,000	$.25\ N^2$
BUBBLE	C	45	190	1,225	4,950	19,900	124,750	499,500	$.5\ N^2$
SORT II	T	17	50	250	830	4,100	20,000	75,000	$.206\ N^{1.84}$
DELAYED	S	5	17	46	90	190	490	990	N
REP'T.	C	45	190	1,225	4,950	19,900	124,750	499,500	$.5\ N^2$
SORT III	T	17	34	130	320	600	1,600	3,700	$1.18\ N^{1.18}$
SHELL-	S	13	34	150	450	930	2,600	5,900	$2\ N^{1.18}$
METZNER	C	31	85	320	900	2,100	5,800	13,000	$4\ N^{1.18}$

times pairs of elements were compared. All values in the table were rounded to two significant digits for clarity, except the number of comparisons in Sorts I and II, which are always exact ($(N^2-N)/2$).

One of the effects of sorting normally distributed numbers is evident in Table I: the number of switches in Sorts I and III is less than half the number of compares by an amount equal to the pairs which were equal. That is, almost half of the compared pairs were right to begin with $(A<B)$; almost half had to be switched $(A>B)$; and some were left alone because they were equal $(A=B)$. For this reason the proportionalities shown may increase slightly when these sorts are used on data with very few equal values.

In both Sort I and Sort II all possible pairs of elements were compared once; in 10 numbers the 45 comparisons are: 9 of #1 with the remaining 9; 8 of #2 with the remaining 8; 7 of #3 with the remaining 7; etc., such that the number of comparisons $C = 9+8+7+6+5+4+3+2+1 = 45$.*

In Sort II the number of switches is always less than the number of elements. This is because in this algorithm a switch is executed only when an element has found its place.

In Sort III many elements must be switched more than once, but far fewer compares are executed. One may consider this algorithm to be intelligent enough, so to speak, that it is aware that if $A<B$ and $B<C$ there is no reason to compare A to C; A must be smaller.

Table I also indicates the approximate quantitative relationships between N and C or S for each of the algorithms. A curvilinear regression analysis (1) was performed on the sort times to determine the equations which would predict the sort times given the number of elements. In each of the equations listed below, T is the time in milliseconds, and N is the number of elements. The coefficient and exponent are given to three significant digits only, as this is empirical evidence.

SORT I: $T = .385\ N^{1.84}$
SORT II: $T = .206\ N^{1.84}$
SORT III: $T = 1.18\ N^{1.18}$

Note that the time-saving with Sort II over Sort I is in the coefficient, and that it is in the exponent with Sort III. Figure 5 is a transposed plot of the data in Figure 1, but this time on log-log paper. It is evident that Sorts I and II have equal slopes (thus equal exponents) and that Sort III has a reduced slope.

One cannot resist adding as Table II some sorting times for very large arrays using these three techniques. Of course, one must have available a great deal of memory to perform some of these sorts; and only under special circumstances and with additional merging algorithms can a programmer use these sorting techniques for large disk or tape sorts. A clear indication of the advantage of Sort III over both Sorts I and II can be calculated using data in Tables I and II. For every tenfold increase in elements to be sorted, there is a seventyfold increase in sort time using I and II, but only a fifteenfold increase using Sort III.

FIGURE V--LOG-LOG TRANSFORMATION OF SORT TIME VS SORT SIZE

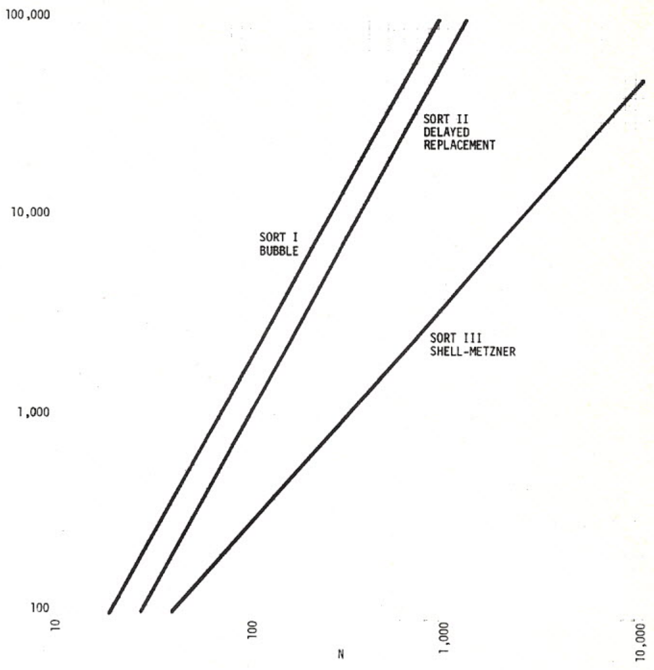


TABLE II--TIMING OF VERY LARGE CORE SORTS

N	Sort I	Sort II	Sort III
10,000	2.5 hrs	1.3 hrs	1 min
100,000	7.1 days	3.8 days	15 min
1,000,000	490 days	260 days	3.9 hrs
10,000,000	93 years	50 years	2.5 days

When this study was started, its purpose was to determine the crossover point at which the Shell-Metzner sort would begin to be more efficient than either of the other two. After all, it does take more coding space, and it does execute more statements given very small sorts. But after dealing with all three of these algorithms, it became more and more obvious that any production core sort code should use Sort III. The only excuse, weak as it is, for using either of the other two would be to teach the basics of sorting algorithms, or of following a flowchart. And under no circumstances should a student ever be taught the bubble sort or the delayed replacement sort without being presented the Shell-Metzner sort as well.

References:

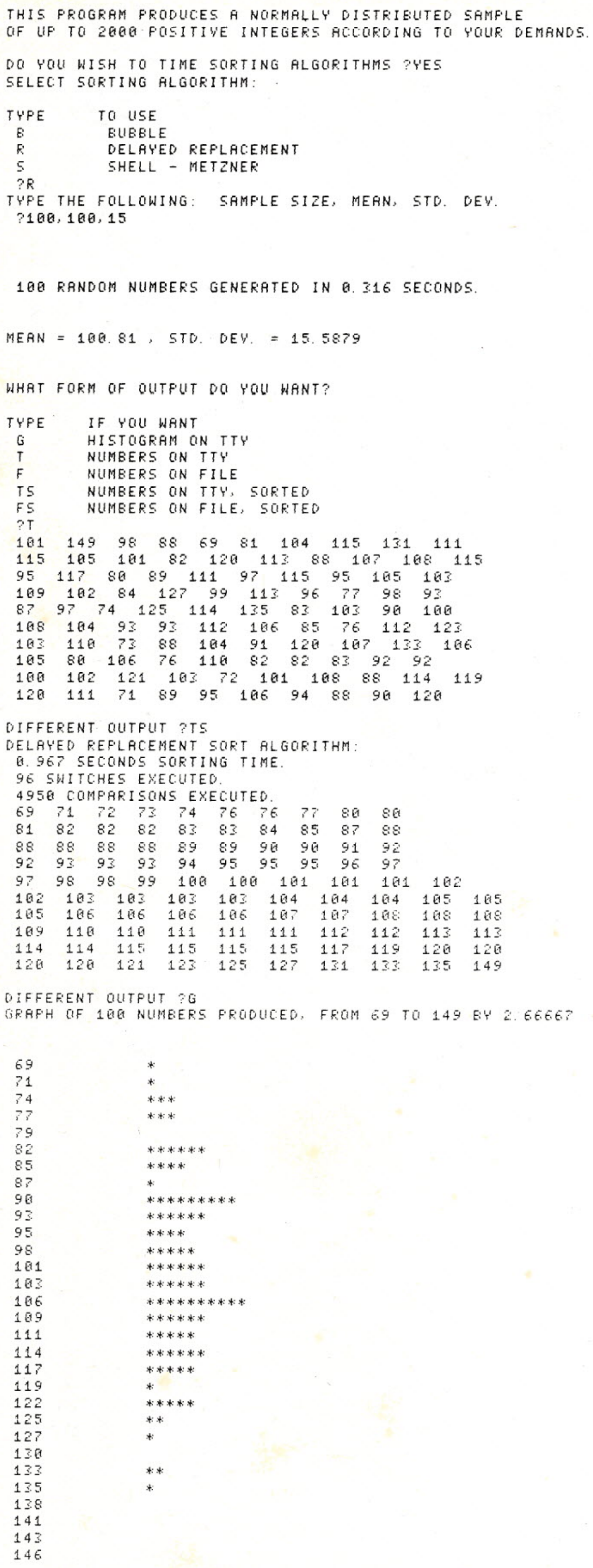
1. Gottlieb, Byron S., Programming with BASIC, McGraw-Hill Book Company, Schaum's Outline Series in Accounting (1975), 175-183.
2. Stuart, Fredric, *FORTRAN Programming,* John Wiley and Sons (New York, 1969), 294-295. From a method published by Marlene Metzner, Pratt and Whitney Aircraft Company. From a method described by D. L. Shell.

John P. Grillo is a convert from chemistry to computing. He was an analytical chemist for Sandia Corporation in Albuquerque, New Mexico for four years. After three years of study at the University of New Mexico for his advanced degree, he taught at Albuquerque Technical-Vocational Institute, then the Computer Information Systems Department at West Texas State University. He is there now as an assistant professor. His research interests are the man-machine interface and CAI

* This is another classic programming problem, the Sum-of-digits. Most teachers force their students to program the brute force sum to teach looping techniques rather than Gauss' elegant Sum = $(N^2+N)/2$.

PROGRAM TO TEST SORT ALGORITHMS

```
10 DIM R2(2000)
20 PRINT "THIS PROGRAM PRODUCES A NORMALLY DISTRIBUTED SAMPLE"
30 PRINT "OF UP TO 2000 POSITIVE INTEGERS ACCORDING TO YOUR DEMANDS."
40 PRINT
50 PRINT "DO YOU WISH TO TIME SORTING ALGORITHMS";
60 INPUT S$
70 IF S$<>"YES" GOTO 150
80 PRINT "SELECT SORTING ALGORITHM:"
90 PRINT
100 PRINT "TYPE      TO USE"
110 PRINT " B         BUBBLE"
120 PRINT " R         DELAYED REPLACEMENT"
130 PRINT " S         SHELL - METZNER"
140 INPUT S$
150 PRINT "TYPE THE FOLLOWING:  SAMPLE SIZE, MEAN, STD. DEV."
160 S2=S4=F=0
170 X1=TIM
180 INPUT Y,M,S
190 IF Y<=2000 GO TO 240
200 PRINT "MAXIMUM SIZE = 2000"
210 GO TO 150
220
230
240 'COMPUTE RANDOM NOS. USING CENTRAL LIMIT THEOREM TECHNIQUE
250 FOR N=1 TO Y
260 R=0
270 FOR J=1 TO 12
280 R=R+RND
290 NEXT J
300 R=M+S*(R-6)
310 R2(N)=INT(R)
320 S2=S2+R2(N)
330 S4=S4+R2(N)*R2(N)
340 NEXT N
350 X2=TIM-X1
360 PRINT
370 PRINT
380 PRINT
390 PRINT Y"RANDOM NUMBERS GENERATED IN"X2"SECONDS."
400 PRINT
410 PRINT
420 M2=S2/Y
430 V2=S4-M2*S2
440 V2=V2/(Y-1)
450 PRINT "MEAN ="M2;
460 PRINT ", STD. DEV. ="SQR(V2)
470 PRINT
480 PRINT
490 PRINT "WHAT FORM OF OUTPUT DO YOU WANT?"
500 PRINT
510 PRINT "TYPE     IF YOU WANT"
520 PRINT " G       HISTOGRAM ON TTY"
530 PRINT " T       NUMBERS ON TTY"
540 PRINT " F       NUMBERS ON FILE"
550 PRINT " TS      NUMBERS ON TTY, SORTED"
560 PRINT " FS      NUMBERS ON FILE, SORTED"
570 INPUT Q$
580 IF LEFT$(Q$,1)<>"F" GOTO 630
590 PRINT "WHAT IS THE NAME OF THE FILE";
600 INPUT F$
610 FILE #1, F$
620 SCRATCH #1
630 IF Q$<>"G" GOTO 680
640 IF F=1 GOTO 660
650 GOSUB 1230
660 GOSUB 1040
670 GOTO 930
680 IF RIGHT$(Q$,1)<>"S" GOTO 710
690 IF F=1 GOTO 710
700 GOSUB 1230
710 IF LEFT$(Q$,1)<>"F" GOTO 800
720 FOR A=1 TO Y BY 10
730 FOR B=A TO A+9
740 IF B>Y GO TO 930
750 PRINT #1,R2(B);
760 NEXT B
770 PRINT #1
780 NEXT A
790 GOTO 930
800 IF LEFT$(Q$,1)="T" GOTO 850
810 IF LEN(Q$)>0 GOTO 830
820 STOP
830 PRINT "IMPROPER OUTPUT CODE; TRY AGAIN"
840 GOTO 470
850 FOR A=1 TO Y BY 10
860 FOR B=A TO A+9
870 IF B>Y GOTO 920
880 PRINT R2(B);
890 NEXT B
900 PRINT
910 NEXT A
920 PRINT
930 PRINT "DIFFERENT OUTPUT";
940 INPUT Q$
950 IF Q$ = "YES" GOTO 510
960 IF Q$<>"NO" GOTO 580
970 PRINT "DO YOU WANT ANOTHER SET OF NUMBERS";
980 INPUT T$
990 IF T$="YES" GO TO 150
1000 STOP
1010
1020
1030 'GRAPHING ROUTINE
1040 L=R2(1)
1050 H=R2(Y)
1060 I=(H-L)/30
1070 PRINT "GRAPH OF"Y"NUMBERS PRODUCED, FROM"L"TO"H"BY"I
1080 PRINT
1090 PRINT
1100 B=1
1110 FOR A=L TO H BY I
1120 PRINT INT(A),
1130 IF R2(B)<R2(B-1) GOTO 1200
1140 IF R2(B)>A GOTO 1180
1150 PRINT "*";
1160 B=B+1
1170 GOTO 1130
1180 PRINT
1190 NEXT A
1200 PRINT
1210 PRINT
1220 RETURN
1230 'SORTING ROUTINE
1240 F=1
1250 X1=TIM
1260 IF S$="R" GOTO 1500
1270 IF S$="B" GOTO 1320
1280 GOTO 1710
1290
1300
1310 'BUBBLE SORT
1320 PRINT "BUBBLE SORT ALGORITHM:"
1330 N7=C7=0
1340 FOF A=1 TO Y-1
1350 FOR B=A+1 TO Y
1360 C7=C7+1
1370 IF R2(A)<R2(B)GOTO 1420
1380 N7=N7+1
1390 T=R2(A)
1400 R2(A)=R2(B)
1410 R2(B)=T
1420 NEXT B,A
1430 X2=TIM-X1
1440 PRINT X2"SECONDS SORTING TIME."
1450 PRINT N7"SWITCHES EXECUTED."
1460 PRINT C7"COMPARISONS EXECUTED."
1470 RETURN
1480
1490
1500      'DELAYED REPLACEMENT SORT
1510      PRINT "DELAYED REPLACEMENT SORT ALGORITHM:"
1520 N7=C7=0
1530 J7=K7=L7=0
1540 L7=L7+1
1550 IF L7=Y GOTO 1430
1560 J7=L7
1570 K7=J7+1
1580 C7=C7+1
1590 IF R2(K7)>R2(J7) GOTO 1610
1600 J7=K7
1610 K7=K7+1
1620 IF K7<=Y GOTO 1580
1630 IF L7=J7 GOTO 1540
1640 N7=N7+1
1650 T=R2(J7)
1660 R2(J7)=R2(L7)
1670 R2(L7)=T
1680 GOTO 1540
1690
1700
1710 'SHELL - METZNER SORT
1720 PRINT "SHELL - METZNER SORT:"
1730 N7=C7=0
1740 M6=Y
1750 M6=INT(M6/2)
1760 IF M6=0 GOTO 1430
1770 K6=Y-M6
1780 J6=1
1790 I6=J6
1800 L6=I6+M6
1810 C7=C7+1
1820 IF R2(I6)<=R2(L6) GOTO 1890
1830 N7=N7+1
1840 T=R2(I6)
1850 R2(I6)=R2(L6)
1860 R2(L6)=T
1870 I6=I6-M6
1880 IF I6>=1 GOTO 1800
1890 J6=J6+1
1900 IF J6>K6 GOTO 1750
1910 GOTO 1790
1920 END

READY
```



```
THIS PROGRAM PRODUCES A NORMALLY DISTRIBUTED SAMPLE
OF UP TO 2000 POSITIVE INTEGERS ACCORDING TO YOUR DEMANDS.

DO YOU WISH TO TIME SORTING ALGORITHMS ?YES
SELECT SORTING ALGORITHM:

TYPE       TO USE
 B          BUBBLE
 R          DELAYED REPLACEMENT
 S          SHELL - METZNER
 ?R
TYPE THE FOLLOWING:  SAMPLE SIZE, MEAN, STD. DEV.
 ?100,100,15



 100 RANDOM NUMBERS GENERATED IN 0.316 SECONDS.


MEAN = 100.81 , STD. DEV. = 15.5879


WHAT FORM OF OUTPUT DO YOU WANT?

TYPE     IF YOU WANT
 G       HISTOGRAM ON TTY
 T       NUMBERS ON TTY
 F       NUMBERS ON FILE
 TS      NUMBERS ON TTY, SORTED
 FS      NUMBERS ON FILE, SORTED
 ?T
 101  149  98  88  69  81  104  115  131  111
 115  105  101  82  120  113  88  107  108  115
 95  117  80  89  111  97  115  95  105  103
 109  102  84  127  99  113  96  77  98  93
 87  97  74  125  114  135  83  103  90  100
 108  104  93  93  112  106  85  76  112  123
 103  110  73  88  104  91  120  107  133  106
 105  80  106  76  110  82  82  83  92  92
 100  102  121  103  72  101  108  88  114  119
 120  111  71  89  95  106  94  88  90  120

DIFFERENT OUTPUT ?TS
DELAYED REPLACEMENT SORT ALGORITHM:
 0.967 SECONDS SORTING TIME.
 96 SWITCHES EXECUTED.
 4950 COMPARISONS EXECUTED.
 69  71  72  73  74  76  76  77  80  80
 81  82  82  82  83  83  84  85  87  88
 88  88  88  88  89  89  90  90  91  92
 92  93  93  93  94  95  95  95  96  97
 97  98  98  99  100  100  101  101  101  102
 102  103  103  103  103  104  104  104  105  105
 105  106  106  106  106  107  107  108  108  108
 109  110  110  111  111  111  112  112  113  113
 114  114  115  115  115  115  117  119  120  120
 120  120  121  123  125  127  131  133  135  149

DIFFERENT OUTPUT ?G
GRAPH OF 100 NUMBERS PRODUCED, FROM 69 TO 149 BY 2.66667


 69           *
 71           *
 74           ***
 77           ***
 79
 82           ******
 85           ****
 87           *
 90           *********
 93           ******
 95           ****
 98           *****
 101          ******
 103          ******
 106          **********
 109          ******
 111          *****
 114          ******
 117          *****
 119          *
 122          *****
 125          **
 127          *
 130
 133          **
 135          *
 138
 141
 143
 146
```

DAYS AND DATES

James Reagan
Mathematics Teacher
Stevenson High School
Sterling Heights, Michigan

Dates become important and remembered because of their importance. You remember your birthday, that perfect date, a confirmation or bar mitzvah, a marriage, divorce, death, birth, or graduation date of yourself or your love. These are personal. Remember the dates? Sure. Remember the day of the week? No? I didn't think so. But now, to take you back in your memory lane to that fond or dreaded day there is a find-the-day-of-the-week formula known as Zeller's Congruence.

If you don't care about your personal past, how knowledgeable are you about your historical past? Try the quiz to see.

An Illustrative Quiz

Provide the date and the day of the week for each of the following events.

1. The stock market crashes beginning the Great Depression.
2. The Second Continental Congress adopts the Declaration of Independence.
3. Japan attacks Pearl Harbor.
4. President Lincoln is assssinated at Ford's Theater.
5. The bombardment of Fort Sumter begins the Civil War.
6. General Custer makes his "last stand" at Little Big Horn.
7. Russia launches Sputnik I, the first artificial satellite.
8. The United States of America drops an atomic bomb on Hiroshima, Japan.
9. President Kennedy is slain by an assassin's bullet in Dallas, Texas.
10. The oceanliner Lusitania is sunk by German U-boat torpedoes killing 1198 persons including 124 Americans.
11. German armies invade Poland starting World War Two.
12. The United States Supreme Court rules in the case of *Brown v Board of Education* that separate schools based upon skin color are inherently unequal.
13. South Korea is invaded by North Korean troops.
14. Richard M. Nixon resigns as President of the United States of America.
15. D-day. Allied troops land in Normandy, France.

Scoring: Count 1 point each for month, day of the month, and year; count 5 points for correct day of the week. There are a possible 8 points for each event with 120 possible points for the quiz. If you scored 0-10 points you are about average; 11-20 points above average; 21-40 points superior; 41-80 points unbelievable; 81-120 points an historical nut — congratulations!!!

For those of you who need help 1) reread history books for the date and 2) utilize Zeller's Congruence to determine the day of the week for any particular date. The formula is:

$$F = (| 2.6m - 0.2 | + k + d + \frac{d}{4} + \frac{c}{4} - 2c) \bmod 7.$$

In this formula, F will have a value 0, 1, 2, 3, 4, 5, or 6; the corresponding day of the week is Sunday, Monday, Tuesday, Wednesday, ..., or Saturday. The modulus 7 can be thought of as the remainder when the value of the parenthetical expression is divided by 7.

The righthand side of the congruence contains the variables described as follows:

k is the day of the month,
c is the number of hundreds in the year,
d is the year in the century, and
m is the month number, but not the layman's month number.
January and February are month numbers 11 and 12 of the preceding year (affecting *d* and possibly *c* described above), March is month number 1, April is month 2, May is 3, ..., and December is month number 10.

The square brackets,| |, indicate that the "greatest integer value" is to be applied to the included expression. A specific example follows.

Example

The date is October 12, 1956. In layman's terms the date is expressed as 10, 12, 1956. For Zeller's Congruence we use m = 8, k = 12, c = 19, and d = 56.

Substituting these values into the right side of the congruence we have

$$F = (| 2.6*8 - 0.2 | + 12 + 56 + \frac{56}{4} + \frac{19}{4} - 2*19) \bmod 7$$

$$= (| 20.8 - 0.2 | + 12 + 56 + | 14 | + | 4.75 | - 38) \bmod 7$$

$$= (20 + 12 + 56 + 14 + 4 - 38) \bmod 7$$

$$= (68) \bmod 7$$

$$= 5 \bmod 7.$$

Thus, we conclude that the day of the week is Friday.

In the application of the formula the following mapping may be a helpful study guide.

LAYMAN'S NOTATION	FORMULA REQUIRES	F VALUE COMPUTED	DAY OF THE WEEK
10-12-1956	8,12,19,56	5	Friday
9-18-1963	7,18,19,63	3	Wednesday
12-25-1972	10,25,19,72	1	Monday
3- 9-1929	1, 9,19,29	6	Saturday
2- 6-1976	12, 6,19,75	5	Friday
1-13-1970	11,13,19,69	2	Tuesday
1- 1-2000	11, 1,19,99	6	Saturday

It might be helpful to understand that the month numbers for the application of the congruence begin with March = 1 and continue to the following February = 12; in this way any leap year day is placed at the end of the formula year.

The First Problem

Write a program that will accept any date in layman's terms and print the corresponding day of the week. The program must provide the translation for the application of the variables used in the congruence. For example, if one types 1,13,1974 the program must translate these values to 11,13,19,73 for the corresponding values of *m, k, c* and *d,* respectively. Using this program you may verify the days of the week for the dates of the Illustrative Quiz.

The Second Problem

Superstitions have developed over the history of man. Many people are superstitious of certain events; those who are not superstitious have some knowledge of the superstitions. Some of the events associated with "bad luck" are: walking under a ladder, having a black cat cross one's path, and breaking a mirror. Perhaps the most well known of all superstitions involves "Black Friday," the description of Friday the Thirteenth.

This year, 1976, has two Friday the Thirteenths; one occurred in February and the other in August. This may be verified by a search of the calendar or by observation of a perpetual calendar.

The second problem becomes one of modifying the program produced to solve the first problem: produce a list of Friday the Thirteenths over a given interval of years. For example, produce a list of Friday the Thirteenths for the years from 1977 to 1980.

The Third Problem

This third problem might be investigated using the computer program produced for the second problem. However, there is also a rigorous mathematical proof of the conjectures motivated by the computer investigation.

The problem is stated in the form of two questions:

1. What is the most number of Friday the Thirteenths in any given year?
2. Is there any year that does not have at least one Friday the Thirteenth?

The Fourth Problem

Some workers are paid bi-weekly, that is they are paid every-other week. The traditional payday is Friday. In a given year there are some months that have 5 Fridays; two of these months occur so that there are 3 paydays in that month, one on each of the first, third and fifth Fridays. The month of February has 4 of each day of the week except in years that are leap years; then one day occurs five times. If that day that occurs five times is Friday, there is a possibility that three paydays may occur in that month.

In what years will February have five Fridays? How often does this occur? If one has bi-weekly pay-periods and one of them does occur on the first Friday of a leap year February beginning on Friday, will the same situation occur again in the worker's lifetime?

Answers To Illustrative Quiz

1. October 24, 1929: Thursday
2. July 4, 1776: Sunday
3. December 7, 1941: Sunday
4. April 14, 1865: Tuesday
5. April 12, 1861: Friday
6. June 25, 1876: Sunday
7. October 4, 1957: Friday
8. August 6, 1945: Monday
9. November 22, 1963: Friday
10. May 7, 1915: Friday
11. September 1, 1939: Friday
12. May 17, 1954: Monday
13. June 25, 1950: Monday
14. August 9, 1974: Friday
15. June 6, 1944: Tuesday

ANSWERS TO COMPUTER LITERACY QUIZ

1. F	7. F	13. 4	19. F	25. F
2. F	8. F	14. T	20. T	26. T
3. 1	9. T	15. T	21. F	27. T
4. T	10. 1	16. 2	22. F	28. 4
5. 5	11. T	17. 3	23. F	29. 4
6. 2	12. 3	18. T	24. 4	30. F

Puzzles and Problems for Fun

JOKER

Four cards, one of each suit, and one each Jack, Queen, King and Ace are laid out in a row.

1. The heart isn't next to the club.
2. No card is next to its immediate senior in rank.
3. The colors of the suits alternate.
4. The king and queen face in opposite directions.
5. The Jack of diamonds is not in the row.

Identify the four cards.

YEAR IN, YEAR OUT

If you take the figures in the year 1974 and multiply them together (1 x 9 x 7 x 4) you end up with the product 252. What is the next date on which this happens?

Similarly 1975 multiplies out to give 315; how long will we have to wait until this happens again?

Games & Puzzles

COMPUTER RECREATIONS

by D. Van Tassel

Chess Programs

Chess is a popular game but not many of us can write a program to play chess. But there are many interesting programs that can be based on chess pieces.

A real simple program is one that reads the row and column of the queen as input, and as output prints a picture of the board with the square the queen is on marked with a Q. Next mark the squares the queen could move to with *'s and mark all other squares with +'s. A queen can move vertically, horizontally, or diagonally as far as desired. For example, a chess-board with a queen in the second row and third column would look as follows:

```
+ * * * + + + +
* * Q * * * * *
+ * * * + + + +
* + * + * + + +
+ + * + + * + +
+ + * + + + * +
+ + * + + + + *
+ + * + + + + +
```

A more interesting program is the Eight Queens problem: Write a program to place 8 queens on a chessboard such that no queen can take any of the others. This means that no two queens may be on the same row, on the same column, or on the same diagonal. This is not a trivial program nor is impossible. I suggest you try it by hand before attempting the program. An elegant solution by Niklaus Wirth is discussed in the book *Structured Programming* by Dahl, Kijkstra, and Hoare, Academic Press.

FILLERUP

Try to arrange the full names of the fifty states into an interlocking pattern, crossword-fashion, minimizing the area of the rectangle into which the completed pattern will fit. All of the state names must be used once and only once. All of the names must be interconnected; that is, no name or group of names may be unconnected from the rest of the names in the completed diagram. Your crossword diagram must be of the kind which uses blacked-out squares. Spaces in state-names are to be ignored.

Games & Puzzles

READING MATTER

A printer uses 1215 characters to number the pages of a book. How many pages are in the book?

Thinkers' Corner

by Layman E. Allen © 1976

WORD PUZZLES

How many of the problems (a) through (f) below can you solve by forming a network of words that have exactly as many letters as the number listed as the GOAL? (Suppose that each symbol below is imprinted on a disc.)

To qualify as a network

(1) all sequences of discs across and down must be words,
(2) the words must have two or more letters and not be proper names,
(3) all of the discs in the REQUIRED column must be used,
(4) as many of the discs in PERMITTED as you wish may be used, and
(5) at most one of the discs in RESOURCES may be used.

Example: The number of letters in the words of the network

```
CAT    is 7:        CAT=3, TO=2, ON=2
 ON                  3 + 2 + 2 = 7
```

The number in the network CAT is 3.

Problem	GOAL	REQUIRED	PERMITTED	RESOURCES
(a)	7	DR	EORT	CEKLMPS
(b)	8	EIZ	EHKO	DENOPRS
(c)	10	CEO	AE MOT	ACKMNQU
(d)	14	EOS	AMPSU	ADNORST
(e)	16	NOW	ANNOTW	ACFGHNP
(f)	21	EITV	ELNNRST	ABEMSRX

Some Suggested Answers (frequently there are others):

```
(a)  R          (c) ONCE        (e) WANT
     O               O                NOON
     S              AT                  W
    RED              E                  N

(b) SHOE        (d) USE         (f)  L
    I               SOAP            SIR
    Z                 R             EVEN
    E                               TEN
                                      T
```

If you enjoy this kind of puzzle, you may like playing ON-WORDS: The Game of Word Structures. Free information about this and other instructional games is available upon request from The Foundation for the Enhancement of Human Intelligence, 1900-W Packard Road, Ann Arbor, MI 48104.

Progression Problems

Charles A. Reeves
Tallahassee, Florida

Last week we grew paramecium in a hay infusion, as described in the experiment from our science book (*Today's Basic Science*). We put some hay in a bucket of tap water, and left it sitting by itself for 7 days. At the end of that time, we had a bucket full of the things.

The book also mentioned that paramecium reproduce by cell division about every 5 hours. Assume that there was only 1 paramecium in the bucket when we started — how many would there be at the end of the 7th day?

For those who want more: Have the computer print the number of paramecium at the end of the 4th, 5th, 6th, and 7th days, all in one run!

On page 194 of *Today's Basic Science,* you will find:
"The female grasshopper is especially adapted for egg-laying. The female lays from 20 to 100 eggs. It lays the eggs in the ground or perhaps in a rotted log. A structure at the tip of the abdomen enables the female to dig a hole in the ground or in rotted wood. This structure is called the 'ovipositer'."

Assume for a moment that you are a scientist, doing an experiment with grasshoppers over a ten-year period. You are applying for a grant from the U. S. government, and so you have to plan how much money you will spend on food, tags, etc. for these animals.

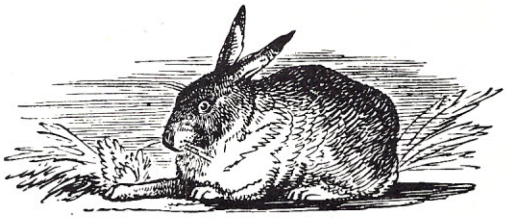
You have to first find out how many grasshoppers you will have in a ten year period (you are starting the experiment with only 1 pair, a male and a female). Have the computer calculate and report to you approximately how many grasshoppers will be born from that one pair. Grasshoppers live only one year, so the females will lay eggs only once in their lives. Assume also that half of those born will be males.

Write a program that you can use to find the average of a given set of numbers. We will use this program to find the class average on tests, and to find the average height and weight of the class. You will want to tell the computer to save this program for future usage.

It takes nature about 500 years to produce 1 inch of topsoil. Many years ago our country had an average depth of almost 9 inches of this good dirt, but now we are down to 6 inches. This type of dirt is necessary, of course, for growing food.

Careless management of our soil causes about 1% per year to erode away, and then it's lost forever. Once we get down to less than 3 inches, it will be impossible to grow crops on a major scale. Have the computer calculate and report to you the year that our country will have less than 3 inches of topsoil, assuming that it continues to erode away at 1% per year.

Will you be alive then? Will your children be alive?



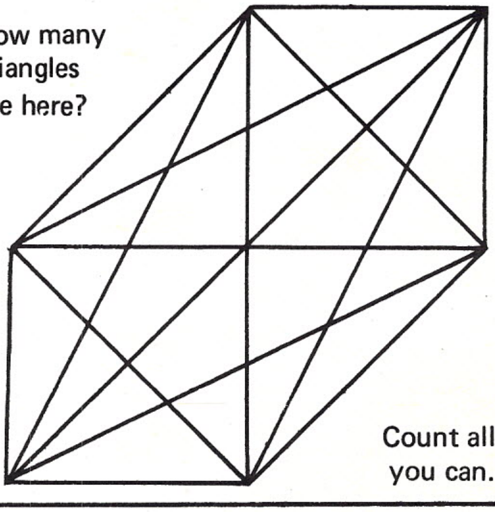
Jack got a pair of bunnies as a New Year's Day present in 1972. This pair became a pair of young rabbits in February, and a pair of adult rabbits in March. A pair of adult rabbits produces a pair of bunnies each month from then on, and this growth cycle continues. The number of pairs of rabbits of each type is provided below, for the first six months:

	J	F	M	A	M	J	
Pairs of bunnies	1	0	1	1	2	3	. . .
Pairs of young	0	1	0	1	1	2	. . .
Pairs of adults	0	0	1	1	2	3	. . .
Total pairs	1	1	2	3	5	8	. . .

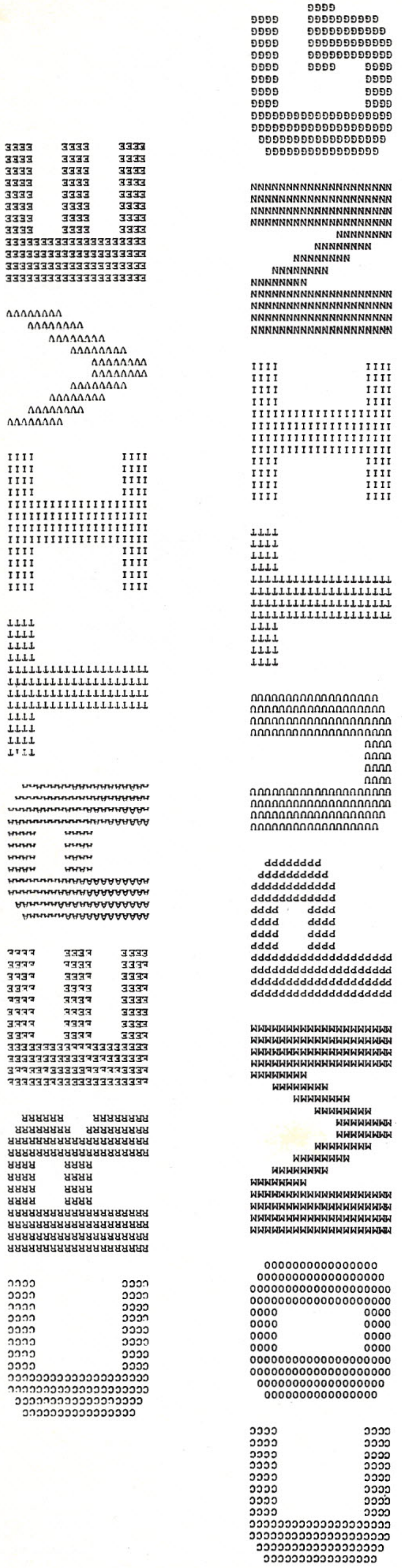
Have the computer tell you the maximum number of rabbits that Jack could have in three years. [Saying the "maximum number" means we are assuming that, of each pair born, one is a male and one a female, and that none of the rabbits die over this period of time.]

TRIANGLES

How many triangles are here?



Count all you can.



POSTER

Bradford Huntress
Hingham, Mass.

The program POSTER is an adaptation of the program *Banner.* It was designed to run under Educomp EDU250 BASIC, but it can easily be adapted to other BASICs as well.

What started out to be a modest attempt merely to adapt the original program to our BASIC ended up as a complete revision now bearing little resemblance to its predecessor. The first problem was finding a simulation for the statement 'change'. I found an EDU250 user-defined function that simulated the ASC( ) function, so the first problem was solved.

Meanwhile, the letters the program would print out were sometimes garbled and altogether plain-looking. They all had square corners with very simple subroutines that controlled their printout. For our system which has less than 4K core per user, the fact that each subroutine performed one and only one function per subroutine, was extremely wasteful. So in my program, I made multi-purpose subroutines that performed two or more functions per subroutine. In doing this I completely changed the shape of all the letters and added a few more characters of my own. All the letters now have rounded corners where appropriate. In my program no two letters except the 'O' and the zero print out the same way. In the former version the '2' and the 'Z' are the same, the 'I' and the '1' are the same, etc. Since it now bears so little resemblance to its forerunner, I decided to call it POSTER instead of Banner.

The software:

Educomp EDU250 BASIC under OS/8 with the following "non-standard" features:

- IF THEN (statement)
- Multiple statements per line; delimiter=':'
- CAT(X$,Y$) concatenates two strings and puts it in the assigned string.
- MID(Z$,X,Y) takes a substring of Z$ starting at X and counting Y characters, and puts it in the assigned string.

Adaptations:

The statement DEF FNA(X)=CAT(" +0",CAT(MID (X,1,1)," "))—992 simulates the ASC( ) function which takes the string argument and finds the ASCII code for it.

To run the program:

Type 'RUN'

The computer will query you for the height, width, and lefthand margin in inches. It will then ask you to type in your message. The computer does the rest.

```
2 !BRADFORD HUNTRESS H.H.S. HINGHAM MASS
200 PRINT"INPUT HEIGHT, WIDTH, & LEFTHAND MARGIN.(ALL IN INCHES)":INPUTL
,R,S
260 PRINT"INPUT MESSAGE":LINPUTB$
261 DEFFNA(X)=CAT(" +0",CAT(MID(X,1,1),"   "))-992
262 S=S*12:A=2*R:GOSUB1500:H$=CAT(H$(L)," "):GOSUB940
270 A=2*R:F=F+1:A2=INT((F/6-INT(F/6))*6+.001):C=A:IFF>B$(0)THEN990
271 IFF/6=INT(F/6)THENA2=6
272 J$=MID(B$(INT((F-1)/6)+1),A2,1):A1=FNA(J$)
277 G$="":FORX=1TOL:G$=CAT(G$,J$):NEXTX
284 FORX=1TOL:G$(X)="":FORY=1TOX:G$(X)=CAT(G$(X),J$):NEXTY
286 NEXTX
290 FORX=1TOA/2:PRINT:NEXTX
294 IFA1=47THENGOSUB880:GOTO270
295 IFA1=39THENG6=1:GOSUB1600:G6=0:GOSUB1600:GOTO270
296 IFA1=33THENG6=2:GOSUB1800:G6=1:GOSUB1800:GOTO270
300 IFA1=46THEN345
302 IFA1=32THEN630
303 IFA1<48THEN270
305 IFA1>57THEN310
306 ONA1-47GOTO500,445,621,625,635,552,615,605,595,585
310 IFA1<65THEN270
320 ONA1-64GOTO350,360,370,380,390,400,410,430,440,450,460,470,480,490,5
00,510,520
331 ONA1-81GOTO530,550,560,570,580,590,600,610,620:GOTO270
345 FORX=1TOA:PRINTTAB(S);G$;G$:NEXTX
349 GOTO 270
350 G6=0:GOSUB1300:A=C/2:GOSUB640:A=C:GOSUB690:A=C/2:GOSUB640:GOSUB1400:
GOTO270
360 GOSUB640:GOSUB650:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
370 G6=1:GOSUB840:A=C/2:GOSUB640:A=C:GOSUB700:GOSUB700:GOTO270
380 GOSUB640:GOSUB700:A=C/2:GOSUB640:G6=0:GOSUB840:GOTO270
390 GOSUB640:GOSUB650:GOSUB650:GOTO270
400 GOSUB640:GOSUB690:GOSUB690:GOTO270
410 G6=1:GOSUB840:A=C/2:GOSUB640:A=C*.75:GOSUB700:A=A/3:GOSUB650:A=2*A
411 GOSUB760:G6=3:GOSUB1800:A=A/2:GOSUB710:GOTO270
430 GOSUB640:GOSUB710:GOSUB640:GOTO270
440 GOSUB700:GOSUB640:GOSUB700:GOTO270
445 GOSUB912:A=C/2:GOSUB700:A=C:GOSUB640:GOSUB740:GOTO270
450 GOSUB645:A=C/2:GOSUB750:A=C:GOSUB740:A=C/2:GOSUB640:G6=1:GOSUB1400:G
OTO270
460 GOSUB640:GOSUB860:GOTO270
470 GOSUB640:GOSUB740:GOSUB740:GOTO 270
480 GOSUB 640:GOSUB890:GOSUB880:GOSUB640:GOTO270
490 GOSUB 640:GOSUB890:GOSUB640:GOTO270
500 A=C/2:G6=1:GOSUB840:GOSUB640:GOSUB700:GOSUB700:GOSUB640:G6=0:GOSUB84
0:GOTO270
510 GOSUB640:GOSUB690:A=C/2:GOSUB790:G6=0:GOSUB1600:GOTO270
520 G6=1:GOSUB840:A=C/2:GOSUB640:A=C*.75:GOSUB700:A=C/4:GOSUB650
521 A=C/2:GOSUB640:A=C/4:G6=0:GOSUB840:GOSUB710:A=C:GOTO270
530 GOSUB640:GOSUB690:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
540 GOSUB640:GOSUB690:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
550 G6=2:GOSUB1800:A=C/2:GOSUB 770:A=C:GOSUB650:A=C/2:GOSUB 760:G6=3
551 GOSUB1800:GOTO270
552 GOSUB770:GOSUB650:A=C/2:GOSUB760:G6=3:GOSUB1800:GOTO270
560 GOSUB780:GOSUB640:GOSUB780:GOTO270
570 G6=1:GOSUB1300:A=C/2:GOSUB640:A=Ç:GOSUB740:A=C/2:GOSUB640:GOSUB1400:
GOTO270
580 GOSUB890:GOSUB880:GOTO270
585 G6=1:GOSUB1600:A=C/2:GOSUB790:A=C:GOSUB690:A=C/2:GOSUB640:G6=0:GOSUB
1400:GOTO270
590 G6=1:GOSUB1300:A=C/2:GOSUB640:A=C:GOSUB880:GOSUB890:A=C/2:GOSUB640
591 GOSUB1400:GOTO270
595 G6=1:GOSUB805:A=C/2:GOSUB640:A=C: GOSUB650:A=C/2:GOSUB640:G6=0
596 GOSUB805:GOTO270
600 GOSUB900:GOTO270
605 GOSUB780:GOSUB780:GOSUB640:GOTO270
610 G6=0:GOSUB960:GOSUB800:G6=1:GOSUB960:GOTO270
615 A=C/2:G6=1:GOSUB840:GOSUB640:GOSUB650:GOSUB650:GOSUB760:G6=3:GOSUB18
00:GOTO270
620 GOSUB920:GOTO270
621 A=C/2:G6=4:GOSUB1800:GOSUB760:A=C:GOSUB650:A=C/2:GOSUB770:G6=1:GOSUB
1800:GOTO270
625 GOSUB700:GOSUB650:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
630 FORX=1TOC*3:PRINT:NEXTX
631 GOTO270
635 GOSUB790:GOSUB710:GOSUB640:GOTO270
640 FORY=1TOA:PRINTTAB(S);:FORX=1TO10:PRINTG$;:NEXTX
641 PRINT:NEXTY
642 RETURN
645 FORX=C/2TO1STEP-1:PRINTTAB(S+X);:FORY=1TO4*L-X:PRINTG$(1);:NEXTY
646 PRINT:NEXTX
647 RETURN
650 IF A<1THENA=1
670 FORX=1TOA:PRINTTAB(S);G$;G$;H$;H$;G$;G$;H$;H$;G$;G$:NEXTX
671 RETURN
690 FORX=1TOA:PRINTTAB(4*L+S);G$;G$;H$;H$;G$;G$:NEXTX
691 RETURN
700 FORX=1TOA:PRINTTAB(S);G$;G$;:PRINTTAB(S+8*L);G$;G$:NEXTX
701 RETURN
710 IF X<1THENA=1
711 FORX=1TOA:PRINTTAB(4*L+S);G$;G$:NEXTX
712 RETURN
740 FORX=1TOA:PRINTTAB(S);G$;G$:NEXTX
746 RETURN
750 FORX=1TOA:PRINTTAB(S);G$;G$;G$;G$:NEXTX
756 RETURN
760 FORY=1TOA:PRINTTAB(S);G$;G$;G$;G$;G$;G$;H$;H$;G$;G$:NEXTY
766 RETURN
770 FORX=1TOA:PRINTTAB(S);G$;G$;H$;H$;G$;G$;G$;G$;G$;G$:NEXTX
776 RETURN
780 FORX=1TOA:PRINTTAB(8*L+S);G$;G$:NEXTX
786 RETURN
790 FORX=1TOA:PRINTTAB(4*L+S);G$;G$;G$;G$;G$;G$:NEXTX
796 RETURN
800 FORX=1TOA:PRINTTAB(S);G$;G$;G$;G$;G$;G$:NEXTX
801 RETURN
805 IFG6=1THENG7=C/2:G9=-1:G8=1:GOTO810
806 G7=1:G9=1:G8=C/2
810 FORX=G7TOG8STEPG9:IFA1=82THENPRINTTAB(S);:GOTO812
811 PRINTTAB(S+X);
812 FORY=1TO2:IFA1=82THENIFY=1THENK9=X:GOTO814
813 K9=2*X
814 FORZ=1TO5*L-K9:PRINTG$(1);:NEXTZ
820 IFX>3THEN950
821 V=X
830 PRINTH$(2*V);" ";:NEXTY
832 PRINT:NEXTX
834 RETURN
840 IFG6=1THENG7=C/2:G8=1:G9=-1:GOTO842
841 G7=1:G8=C/2:G9=1
842 FORX=G7TOG8STEPG9:PRINTTAB(S+X);:FORY=1TO10*L-2*X:PRINTG$(1);:NEXTY
843 PRINT:NEXTX
844 RETURN
860 IFR>LTHENG6=L*4:G7=1:GOTO865
861 G7=INT(L/R+.5):G6=C*2*L/R
865 FORX=1TOG6STEPG7:PRINTTAB(S+4*L-X);G$;G$;TAB(S+4*L+X);G$;G$:NEXTX
871 RETURN
880 FORX=0TO6*LSTEP 6*L/C:PRINTTAB(X+S);G$;G$;G$;G$:NEXTX
881 RETURN
890 FORX=6*LTO0STEP-6*L/C:PRINTTAB(X+S);G$;G$;G$;G$:NEXTX
891 RETURN
900 FORX=0TO7*LSTEP(14*L)/(C*6):PRINTTAB(X+S);G$;G$;G$;
910 PRINTTAB(7*L-X+S);G$;G$;G$:NEXTX
911 RETURN
912 FORX=1TOC/2:PRINTTAB(S);G$;G$;:PRINTTAB(S+8*L);G$;:FORY=1TOX:PRINTG$
(1);:NEXTY
913 PRINT:NEXTX
914 RETURN
920 FORX=0TO7*LSTEP(7*L)/(C*3):PRINTTAB(S);G$;G$;TAB(X+S);G$;G$;G$;
930 PRINTTAB(8*L+S);G$;G$:NEXTX
931 RETURN
940 FORX=1TOA*3:PRINT:NEXTX
941 RETURN
950 FORW=1TOX-3:PRINT"  ";:NEXTW
951 V=3:GOTO 830
960 IFG6=1THENG7=4*L:G8=7*L:G9=4*L/C:GOTO962
961 G7=7*L:G8=4*L:G9=FIX(-4*L/C)
962 FORX=G7TOG8STEPG9:PRINTTAB(X+S);G$;G$;G$:NEXTX
963 RETURN
980 DATA""," ","  ","   ","    ","     "
990 FORX=1TOC*3:PRINT:NEXTX
1000 STOP
1300 FORX=C/2TO1STEP-1:IFG6=1THENPRINTTAB(S+X);:GOTO1304
1302 PRINTTAB(S);
1304 FORY=1TO10*L-X:PRINTG$(1);:NEXTY
1305 PRINT:NEXTX
1306 RETURN
1400 FORX=1TOC/2:IFG6=1THENPRINTTAB(S+X);:GOTO1403
1402 PRINTTAB(S);
1403 FORY=1TO10*L-X:PRINTG$(1);:NEXTY
1404 PRINT:NEXTX
1405 RETURN
1500 RESTORE:FORJ8=1TO6:READH$(J8):NEXTJ8
1501 RETURN
1600 IFG6=0THENG7=1:G8=C/2:G9=1:GOTO1602
1601 G7=C/2:G8=1:G9=-1
1602 FORX=G7TOG8STEPG9:PRINTTAB(4*L+X+S);:FORY=1TO6*L-2*X:PRINTG$(1);:NE
XTY
1603 PRINT:NEXTX
1604 RETURN
1800 IFG6/2=INT(G6/2)THENG7=C/2:G8=1:G9=-1:GOTO1812
1801 G7=1:G8=C/2:G9=1
1812 FORX=G7TOG8STEPG9:IFG6<=2THENG4=0:G5=4*L:GOSUB1820:GOSUB1830:GOTO18
60
1814 G4=8*L:G5=0:GOSUB1830:GOSUB1820:GOTO1860
1820 PRINTTAB(S+G4);G$;G$;:RETURN
1830 IFG6=4THENIFA1=50THENPRINTTAB(S);G$(X);
1840 PRINTTAB(S+G5+X);:FORY=1TO6*L-2*X:PRINTG$(1);:NEXT Y
1841 IFA1=71THENFORY=1TOX:PRINTG$(1);:NEXTY
1842 RETURN
1860 PRINT:NEXTX
1861 RETURN
```

Another new game from Creative Computing . . . .

LEM

Author: Unknown
Modified by: Bill Cotter, Pittsfield, Mass.
Language: BASIC (Honeywell 600/6000)
Description: The user is put at the controls of yet another Lunar Module. The first task is to pick the initial conditions—speed, etc. This lets the user progress in a learning fashion; there is no random factoring involved.

Factors to be considered:

(1) Landing speed—land harder than 3 meters/sec and that's it.
(2) Moving too fast over the terrain causes you to flip when you land.
(3) Your engines will blow up if used to the limit.

Suggestions: Improve lines 2080-2250 (the landing plot).

PROGRAM LISTING

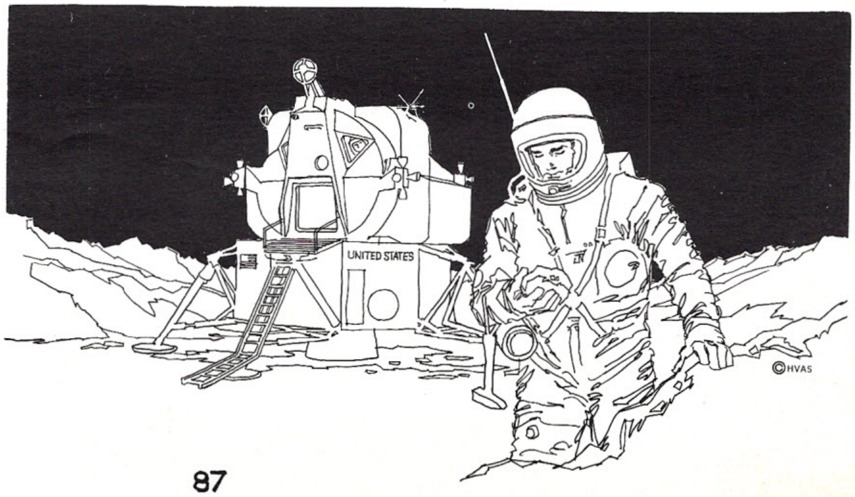
```
100 PRINT "THIS IS A LUNAR LANDING PROGRAM. "
110 PRINT "DO YOU WISH INSTRUCTIONS";
120 INPUT V$
130 IF V$="NO" GOTO 330
140 PRINT "THIS IS THE LANDING ZONE."
150 PRINT TAB(10);"^";TAB(12); "Y AXIS (+)"
160 FOR I=1 TO 4
170 PRINT TAB(10);"!"
180 NEXT I
190 PRINT "----------+---------> X AXIS (+)"
200 FOR I=1 TO 5
210 PRINT TAB(10);"!"
220 NEXT I
230 PRINT "THE POSITIVE Z AXIS IS OUT OF THE PAPER"
240 PRINT "YOU ARE TRYING TO 'LAND ON THE CROSS."
250 PRINT "YOU HAVE CONTROL OF YOUR VERTICAL(Z), HORIZONTAL(Y),"
260 PRINT "AND TRANSVERSE(X) VELOCITIES. YOU ALSO HAVE THE "
270 PRINT "ADDITIONAL ABILITY TO CONTROL THE LENGTH OF TIME OF "
280 PRINT "BURN. YOU WILL SUPPLY ALL INITIAL DATA."
290 PRINT "ALL UNITS ARE METRIC."
300 PRINT
310 PRINT "REMEMBER IF YOU RUN OUT OF FUEL THATS IT."
320 PRINT
330 PRINT "WHAT IS THE INITIAL ALTITUDE";
340 INPUT A3
350 PRINT "WHAT IS THE INITIAL VERTICAL VELOCITY (DOWN IS +)";
360 INPUT V6
370 PRINT "WHAT IS THE DISTANCE Y";
380 INPUT D4
390 PRINT "WHAT IS Y VELOCITY";
400 INPUT V4
410 PRINT "WHAT IS THE DISTANCE X";
420 INPUT D5
430 PRINT "WHAT IS THE X VELOCITY";
440 INPUT V5
450 PRINT "WHAT IS THE MAXIMUM BURN RATE";
460 INPUT M
470 PRINT "WHAT IS YOUR FUEL CAPACITY";
480 INPUT F3
490 PRINT "WHAT IS THE GRAVITATIONAL CONSTANT";
500 INPUT G
510 PRINT "WHAT IS THE NAME OF YOUR SHIP";
520 INPUT N1$
530 PRINT
540 PRINT  "CONTROL TO "N1$" COMMENCE LANDING."
550 GO SUB 1900
560 LET V=V6
570 LET V1=V4
580 LET V2=V5
590 LET T=0
600 LET F=F3
610 LET A=A3
620 LET D1=D4
630 LET D2=D5
640 LET T3=T3+T
650 GOSUB 2080
660 PRINT "TIME ="T3"SEC."
670 PRINT "ALT="A" METERS V="V" METERS/SEC"
680 PRINT "DIST. X="D2"METERS. V="V2"METERS/SEC"
690 PRINT "DIST. Y="D1"METERS. V="V1"METERS/SEC"
700 PRINT "FUEL="F"UNITS"
710 PRINT "TIME INTERVAL";
720 INPUT T
730 PRINT "VERTICAL(Z) BURN";
740 INPUT B
750 PRINT "TRANSVERSE(X) BURN";
760 INPUT B2
770 LET B2=-B2
780 PRINT "HORIZONTAL (Y) BURN";
790 INPUT B1
800 LET B1=-B1
810 PRINT
820 GOTO 1340
830 LET F1=F
840 LET A1=A
850 IF F<=0 GOTO 1190
860 LET A=A-V*T-((G-B)*T)/2
870 IF A<=0 GOTO 910
880 LET V=V+(G-B)*T
890 GO SUB 2010
900 GO TO 640
910 LET G1=G-B
920 LET T1=((-2*V)+SQR(ABS(4*V^2+8*G*A1)))/(2*G)
930 LET V=V+(G1*T1)
940 GOSUB 2010
950 LET T4=T3+T1
960 IF V<3 GOTO 990
970 PRINT "AT T="T4" THE "N1$" CRASHED WITH A DESCENT ";
980 GOTO 1000
990 PRINT "AT T="T4"THE "N1$" LANDED WITH A DESCENT ";
1000 PRINT "VELOCITY OF"V"M/SEC"
1010 LET R=SQR(V1^2+V2^2)
1020 IF R>5 GOTO 1050
1030 PRINT "WITH A HORIZONTAL VELOCITY OF "R"M/SEC"
1040 GOTO 1080
1050 LET P9=1
1060 PRINT "AND FLIPPED OVER WITH A HORIZONTAL VELOCITY OF"R"M/SEC"
1070 LET P9=1
1080 PRINT "THE LANDING POINT WAS AT ("D2","D1")"
1090 LET D=SQR(D1^2+D2^2)
1100 IF R<=5 GOTO 1140
1110 PRINT D"METERS FROM THE LANDING SITE."
1120 GOTO 1790
1130 IF P9=1 GOTO 1790
1140 IF D>100 GOTO 1170
1150 PRINV "BEAUTIFUL "N1$" YOU WERE "D"METERS FROM THE LANDING SITE"
1160 GOTO 1790
1170 PRINT "GOOD LANDING "N1$". BUT YOU WERE "D"METERS OFF"
1180 GOTO 1790
1190 LET T=F1/(ABS(B)+ABS(B1)+ABS(B2))
1200 LET A=A-(V*((G-B)/2))
1210 LET V=V+(G-B)
1220 LET T1=((-2*V)+SQR(4*V^A+8*A1*G))/(2*G)
1230 LET V=V+G*T1
1240 GOSUB 2010
1250 LET D1=D1+V1*(T1-T)
1260 LET D2=D2+V2*(T1-T)
1270 IF V<3 GOTO 950
1280 LET T4=T3+T1
1290 PRINT "THE "N1$;N2" CRASHED AT T="T4"SEC AT THE POINT ("D2","D1")"
1300 LET R=SQR(V1^2+V2^2)
1310 PRINT "WITH A DOWNWARD VELOCITY "V"AND A FORWARD VELOCITY"R
1320 PRINT "CRASH DUE TO PILOT ERROR (THE IDIOT RAN OUT OF FUEL)"
1330 GOTO 1790
1340 IF ABS(B)<=M GOTO1400
1350 IF B<0 GOTO 1380
1360 LET B=M
1370 GOTO 1390
1380 LET B=-M
1390 LET Z=Z+1
1400 IF ABS(B1)<=M GOTO 1460
1410 IF B1<0 GOTO 1440
1420 LET B1=M
1430 GOTO 1450
1440 LET B1=-M
1450 LET Z1=Z1+1
1460 IF ABS(B2)<=M GOTO 1520
1470 IF B2<0 GOTO 1500
1480 LET B2=M
1490 GOTO 1510
1500 LET B2=-M
1510 LET Z2=Z2+1
1520 LET F=F-((ABS(B)+ABS(B1)+ABS(B2))*T)
1530 IF Z=1 GOTO 1600
1540 IF Z=2 GOTO 1780
1550 IF Z1=1 GOTO 1660
1560 IF Z1=2 GOTO 1780
1570 IF Z2=1 GOTO 1720
1580 IF Z2=2 GOTO 1780
1590 GOTO 1770
1600 IF B=0 GOTO 1550
1610 LET E=E-1
1620 IF E=0 GOTO 1780
1630 GOSUB 1880
1640 PRINT "VERTICAL ENGINE WILL BLOW IN"E" BURNS"
1650 GOTO 1550
1660 IF B1=0 GOTO 1570
1670 LET E1=E1-1
1680 IF E1=0 GOTO 1780
1690 GOSUB 1880
1700 PRINT "HORIZONTAL ENGINE WILL BLOW IN"E1" BURNS"
1710 GOTO 1570
1720  IF B2=0 GOTO 1770
1730 LET E2=E2-1
1740 IF E2=0 GOTO 1780
1750 GOSUB 1880
1760 PRINT "TRANSVERSE ENGINE WILL BLOW IN"E2" BURNS"
1770 GOTO 840
```

(MORE)

```
1780 GOSUB 1990
1790 PRINT
1800 PRINT "WOULD YOU LIKE TO TRY TO CRASH IT AGAIN STUPID";
1810 INPUT V$
1820 IF V$="NO"GOTO 1870
1830 PRINT "SAME INITIAL VALUES";
1840 INPUT V$
1850 IF V$="YES" GOTO 530
1860 GOTO 320
1870 STOP
1880 PRINT "  ";
1890 RETURN
1900 LET T3=0
1910 LET Z=0
1920 LET Z1=0
1930 LET Z2=0
1940 LET E=11
1950 LET E1=11
1960 LET E2=11
1970 LET P9=0
1980 RETURN
1990 PRINT "BANG!!!!!!!!!!"
2000 RETURN
2010 LET V7=V1-B1*T
2020 LET D1=D1+(V1*T)+(T*(V1-V7))/2
2030 LET V8=V2-B2*T
2040 LET D2=D2+(V2*T)+(T*(V2-V8))/2
2050 LETV1=V7
2060 LET V2=V8
2070 RETURN
2080 IF A>100 GOTO 2250
2090 IF D2>100 GOTO 2250
2100 IF D1>100 GOTO 2250
2110 LET Q2=INT(D2/10)+10
2120 LET Q1=-INT(D1/20)+5
2130 PRINT "T="T3
2140 FOR I=1 TO 11
2150 IF I<>QJ GOTO 2210
2160 IF I=6 GOTO 2190
2170 PRINT TAB(10);"!";TAB(Q2);"X ALTITUDE="A
2180 GOTO 2240
2190 PRINT "----------+----------";TAB(Q2);"X ALTITUDE=";A
2200 GOTO 2240
2210 IF I=6 GOTO 2230
2220 GOTO 2240
2230 PRINT "----------+----------"
2240 NEXT I
2250 RETURN
2260 END
```

SAMPLE RUN

```
THIS IS A LUNAR LANDING PROGRAM.
DO YOU WISH INSTRUCTIONS ?NO
WHAT IS THE INITIAL ALTITUDE ?600
WHAT IS THE INITIAL VERTICAL VELOCITY (DOWN IS +) ?25
WHAT IS THE DISTANCE Y ?200
WHAT IS Y VELOCITY ?-10
WHAT IS THE DISTANCE X ?25
WHAT IS THE X VELOCITY ?1
WHAT IS THE MAXIMUM BURN RATE ?50
WHAT IS YOUR FUEL CAPACITY ?1200
WHAT IS THE GRAVITATIONAL CONSTANT ?.6
WHAT IS THE NAME OF YOUR SHIP ?AQUARIUS

CONTROL TO AQUARIUS COMMENCE LANDING.
TIME =    0 SEC.
ALT=  600  METERS V=   25  METERS/SEC
DIST. X=   25 METERS. V=    1 METERS/SEC
DIST. Y=  200 METERS. V=  -10 METERS/SEC
FUEL= 1200 UNITS
TIME INTERVAL ?5
VERTICAL(Z) BURN ?0
TRANSVERSE(X) BURN ?-.2
HORIZONTAL (Y) BURN ?0

TIME =    5 SEC.
ALT=   473.5  METERS V=   28  METERS/SEC
DIST. X= 32.5 METERS. V=    0 METERS/SEC
DIST. Y=  150 METERS. V=  -10 METERS/SEC
FUEL= 1199 UNITS
TIME INTERVAL ?10
VERTICAL(Z) BURN ?0
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

TIME =   15 SEC.
ALT=   190.5  METERS V=   34  METERS/SEC
DIST. X= 32.5 METERS. V=    0 METERS/SEC
DIST. Y=   50 METERS. V=  -10 METERS/SEC
FUEL= 1199 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?10
TRANSVERSE(X) BURN ?-.2
HORIZONTAL (Y) BURN ?2

TIME =   17 SEC.
ALT=   131.9  METERS V= 15.2  METERS/SEC
DIST. X= 32.9 METERS. V=  -.4 METERS/SEC
DIST. Y=   26 METERS. V=   -6 METERS/SEC
FUEL=  1174.6 UNITS
TIME INTERVAL ?4
VERTICAL(Z) BURN ?33
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?1.5

T=   21
----------+----------X ALTITUDE= 75.9
TIME =   21 SEC.
ALT= 75.9  METERS V=   5.599999  METERS/SEC
DIST. X= 31.3 METERS. V=  -.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1156.6 UNITS
TIME INTERVAL ?10
VERTICAL(Z) BURN ?.5
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

T=   31
----------+----------X ALTITUDE=   19.40001
TIME =   31 SEC.
ALT=   19.40001  METERS V=   6.599999  METERS/SEC
DIST. X= 27.3 METERS. V=  -.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1151.6 UNITS
TIME INTERVAL ?1
VERTICAL(Z) BURN ?5
TRANSVERSE(X) BURN ?-1
HORIZONTAL (Y) BURN ?0

T=   32
----------+----------X ALTITUDE=   15.00001
TIME =   32 SEC.
ALT=   15.00001  METERS V=   2.199999  METERS/SEC
DIST. X= 27.4 METERS. V= -1.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1145.6 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.5
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

T=   34
----------+----------X ALTITUDE=   10.50001
TIME =   34 SEC.
ALT=   10.50001  METERS V=   2.399999  METERS/SEC
DIST. X= 24.6 METERS. V= -1.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1144.6 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.7
TRANSVERSE(X) BURN ?.7
HORIZONTAL (Y) BURN ?0

T=   36
----------+----------X ALTITUDE=   5.800013
TIME =   36 SEC.
ALT=   5.800013  METERS V=   2.199999  METERS/SEC
DIST. X= 20.4 METERS. V=-1.49012e-08    METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1141.8 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.8
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

T=   38
----------+----------X ALTITUDE=   1.600014
TIME =   38 SEC.
ALT=   1.600014  METERS V=   1.799999  METERS/SEC
DIST. X= 20.4 METERS. V=-1.49012e-08    METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1140.2 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.7
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

AT T=   38.78595 THE AQUARIUS LANDED WITH A DESCENT VELOCITY OF
   1.721405 M/SEC
WITH A HORIZONTAL VELOCITY OF  6.14391e-08    M/SEC
THE LANDING POINT WAS AT ( 20.4 ,  -10 )
BEAUTIFUL AQUARIUS YOU WERE    22.71915 METERS FROM THE LANDING SITE

WOULD YOU LIKE TO TRY TO CRASH IT AGAIN STUPID ?NO
```



Another new game from Creative Computing . . . .

TWO-TO-TEN

Two-to-Ten is a game of chance played with a special deck of cards with only the cards 2 - 10. The game is similar to blackjack in that you are drawing cards and trying to come as close as possible to a goal number (chosen at random before each round) without going over it. You must come within a certain number of points of the goal number determined by a "lucky-limit" card. The catch to the game is that you are not given the exact value of the goal number but rather a clue that is only within 15% of the goal.

Can you think of a way to make Two-to-Ten more interesting? Perhaps playing it against the computer as an opponent? Let's hear your ideas!

I'm embarrassed to say that I don't remember who originally gave me Two-to-Ten, but if the author will drop a line, I'll credit him or her in the next issue. — DHA

SAMPLE RUN

```
RUNNH

WELCOME TO THE GAME TWO-TO-TEN.  THE NAME COMES FROM THE
SPECIAL "DECK OF CARDS" USED.  THERE ARE NO FACE CARDS - ONLY
THE CARDS 2-10.  THIS GAME IS EASY AND FUN TO PLAY IF YOU
UNDERSTAND WHAT YOU ARE DOING SO READ THE INSTRUCTIONS
CAREFULLY.
AT THE START OF THE GAME YOU BET ON WINNING.  TYPE IN ANY
NUMBER BETWEEN 0 AND 200.  RSTS THEN PICKS A RANDOM NUMBER
YOU ARE TO REACH BY THE SUM TOTAL OF MORE CARDS CHOSEN.
BECAUSE OF THE RARE CHANCE OF YOU GETTING TO THAT NUMBER
EXACTLY, YOU ARE GIVEN AN ALLOWANCE CARD.  THE OBJECT OF
THE GAME IS TO GET THE TOTAL OF CARDS WITHIN THE MYSTRY
NUMBER WITHOUT GOING OVER.
YOU ARE GIVEN A HINT AS TO WHAT THE NUMBER IS.  THIS IS NOT
THE EXACT NUMBER ONLY ONE CLOSE.  ALL YOU DO IN THIS GAME IS
DECIDE WHEN TO STOP.  AT THIS POINT YOUR TOTAL IS COMPARED
WITH THE NUMBER AND YOUR WINNINGS ARE DETERMINED.

GOOD LUCK!


PLACE YOUR BET . . . YOU HAVE $ 200 TO SPEND.? 50

YOUR "LUCKY-LIMIT" CARD IS A 5
YOU MUST COME WITHIN 5 WITHOUT GOING OVER TO WIN.

HERE WE GO!


CARD # 1 IS A 10 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 10 .  DO YOU WANT TO CONTINUE? YES


CARD # 2 IS A 4 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 14 .  DO YOU WANT TO CONTINUE? YES


CARD # 3 IS A 7 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 21 .  DO YOU WANT TO CONTINUE? YES


CARD # 4 IS A 6 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 27 .  DO YOU WANT TO CONTINUE? YES


CARD # 5 IS A 9 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 36 .  DO YOU WANT TO CONTINUE? YES


CARD # 6 IS A 5 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 41 .  DO YOU WANT TO CONTINUE? YES


CARD # 7 IS A 5 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 46 .  DO YOU WANT TO CONTINUE? YES


CARD # 8 IS A 4 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 50 .  DO YOU WANT TO CONTINUE? YES


CARD # 9 IS A 9 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 59 .  DO YOU WANT TO CONTINUE? NO

YOU WIN !  THE NUMBER WAS 61 .  YOUR GUESS TOTAL
WAS 59 WITHIN  YOUR LIMIT CARD.
YOU NUW HAVE $ 250 IN CASH TO BET IN THE NEXT GAME!
WOULD YOU LIKE TO PLAY THAT NEXT GAME? NO

HOPE YOU HAD FUN

READY
```

PROGRAM LISTING

```
50 PRINT \PRINT \PRINT
60 PRINT "WELCOME TO THE GAME TWO-TO-TEN.  THE NAME COMES FROM THE"
70 PRINT 'SPECIAL "DECK OF CARDS" USED.  THERE ARE NO FACE CARDS - ONLY'
80 PRINT "THE CARDS 2-10.  THIS GAME IS EASY AND FUN TO PLAY IF YOU"
90 PRINT "UNDERSTAND WHAT YOU ARE DOING SO READ THE INSTRUCTIONS"
100 PRINT "CAREFULLY."
110 PRINT "AT THE START OF THE GAME YOU BET ON WINNING.  TYPE IN ANY"
120 PRINT "NUMBER BETWEEN 0 AND 200.  RSTS THEN PICKS A RANDOM NUMBER
130 PRINT "YOU ARE TO REACH BY THE SUM TOTAL OF MORE CARDS CHOSEN."
140 PRINT "BECAUSE OF THE RARE CHANCE OF YOU GETTING TO THAT NUMBER"
150 PRINT "EXACTLY, YOU ARE GIVEN AN ALLOWANCE CARD.  THE OBJECT OF"
160 PRINT "THE GAME IS TO GET THE TOTAL OF CARDS WITHIN THE MYSTRY
170 PRINT "NUMBER WITHOUT GOING OVER."
180 PRINT "YOU ARE GIVEN A HINT AS TO WHAT THE NUMBER IS.  THIS IS NOT"
185 PRINT "THE EXACT NUMBER ONLY ONE CLOSE.  ALL YOU DO IN THIS GAME IS"
190 PRINT "DECIDE WHEN TO STOP.  AT THIS POINT YOUR TOTAL IS COMPARED"
195 PRINT "WITH THE NUMBER AND YOUR WINNINGS ARE DETERMINED."
197 PRINT\PRINT"GOOD LUCK!"\PRINT\PRINT\PRINT
199 M=200
200 RANDOMIZE
210 D=0\T=0
215 O=INT(10*RND(0))+25
220 N=INT(O*RND(0))+O
230 R=(INT(15*RND(0))+1)/100
250 S=INT(2*RND(0)+1)
260 IF S=1 THEN E=INT(N-(N*R))
265 GOTO 280
270 E=INT(N+(N*R))
280 A=INT(9*RND(0)+2)
285 PRINT "PLACE YOUR BET . . . YOU HAVE $"M"TO SPEND.";\INPUT B\PRINT
290 IF B>M THEN 295 ELSE 300
295 PRINT "YOU CAN'T BET MORE THAN YOU'VE GOT!"\GOTO 250
300 PRINT 'YOUR "LUCKY-LIMIT" CARD IS A'A
310 PRINT "YOU MUST COME WITHIN"A"WITHOUT GOING OVER TO WIN."
320 PRINT \PRINT "HERE WE GO!"
330 PRINT \PRINT
340 D=D+1
350 C=INT(9*RND(0)+2)
360 PRINT "CARD #"D"IS A"C".  YOU ARE TRYING TO COME NEAR"E
365 T=T+C
370 IF T>N THEN 375 ELSE 380
375 PRINT "YOUR TOTAL IS OVER THE NUMBER"N".  AN AUTOMATIC LOSS!"
377 GOTO 570
380 PRINT "TOUR TOTAL IS"T".  DO YOU WANT TO CONTINUE"; \INPUT Q$\PRINT
390 IF LEFT(Q$,1)="Y" THEN 330
410 IF T>=N-A AND T<=N THEN 500 ELSE 550
500 PRINT "YOU WIN !  THE NUMBER WAS"N".  YOUR GUESS TOTAL WAS"T"WITHIN"
510 PRINT "YOUR LIMIT CARD."
520 M=M+B
540 GOTO 600
550 PRINT "YOU BLEW IT!  THE NUMBER WAS"N", OUTSIDE YOUR ALLOWANCE BY";
560 PRINT (N-A)-T \PRINT
570 M=M-B
600 PRINT "YOU NOW HAVE $"M"IN CASH TO BET IN THE NEXT GAME!"
610 INPUT "WOULD YOU LIKE TO PLAY THAT NEXT GAME";Q$\PRINT
620 IF LEFT(Q$,1)="Y"THEN 200
630 PRINT "HOPE YOU HAD FUN"
999 END
```

Creative Computing Feature Review . . .

by Sema Marks

Learning Alternatives In U.S. Education: Where Student And Computer Meet. Beverly Hunter, Carol S. Kastner, Martin L. Rubin, and Robert J. Seidel. Educational Technology Publications, Englewood Cliffs, New Jersey 07632. 1975. xvi + 398 pp., $14.95.

Every few years another major study appears on the future of computers in education. There was the Rosser Report of the National Academy of Sciences in 1966;[1] the Pierce Report of the President's Science Advisory Committee in 1967;[2] the Oettinger and Marks report of the Harvard Program on Technology and Society in 1969;[3] the Tickton report of the Commission on Instructional Technology in 1970;[4] the Levien report of a study for the Carnegie Commission on Higher Education and the Rand Corporation in 1972;[5] *The Fourth Revolution*, a report and recommendations by the Carnegie Commission on Higher Education, also in 1972;[6] and now a report by Beverly Hunter, Carol S. Kastner, Martin L. Rubin and Robert J. Seidel in 1975.

The latest in this series, published as *Learning Alternatives in U.S. Education: Where Student and Computer Meet*, is based on a project performed at the Human Resources Research Organization (HumRRO) under a grant from the National Science Foundation. The purpose of the project was "to study development and dissemination of computer-based learning materials in the U.S. and to identify approaches for achieving beneficial, nationwide use of computers in education."

In the decade since the Rosser Report little has changed in the classroom, but a great deal has changed outside of it. The use of computers has grown substantially and people's attitudes towards them have changed accordingly. We have moved from what Levien calls the "parochial era," during which computer usage was expensive and justified for only a narrow class of numerical and clerical tasks, to what he calls "a universal era," in which computers are economically accessible for a wide class of new and previously infeasible applications.[7]

Before too long computers will be accessible to everyone. Today over 13 million people in the U.S. alone own pocket calculators, perhaps the first sign of free and easy access to computer power for all.[8] Millions more are coming into *direct* contact with computers through point-of-sale terminals located in stores, supermarkets, airline reservation counters and offtrack betting parlors; credit cards; cash machines; cash registers; digital watches; toys and games; electronic mail; and an ever-increasing number of terminals located in schools and colleges, libraries, museums, and store-front and community computing centers. It is even possible today to build your own computer with 4K memory, cassette operating system, alpha-numeric keyboard, and extended Basic for less than $500. By 1985, predicts, F.G. "Buck" Rodgers, Corporate Vice President for Marketing at IBM, one out of every 100 homes will have a computer terminal. Some people think that he is a stodgy pessimist!

It is the *accessibility* to computer power, brought about by greatly reduced costs and high demand, that is the key to its future use. Major changes accompany the use of any new medium as it passes from the hands of the few to the hands of the many, and the history of technology shows that it is next to impossible to predict when a medium is in the hands of the few how it will be used in the hands of the many. Yet the authors of *Learning Alternatives* content themselves with describing the present and simply extrapolating from today to tomorrow. They paint the picture within the frame but fail to consider the changing framework in which it's all happening.

But first for the good news. The authors do a superlative job of telling us where we are. The book is an invaluable source of information about what computer-based learning materials are available today, where to find them, and what you can do with them. The set of references (all twenty-three pages of them) and sources of information listed in the appendices alone are worth the price of the book, and the analyses offered by the authors in the first four chapters are clear and insightful.

The first two chapters, beautifully written by Hunter and Seidel, point out the diversity of purposes and activities which can be served by technology, and the interdependence of social goals, values, educational reform and technology. They provide many useful distinctions such as the difference between an educational innovation designed to be used within the current educational structure and one intended to reform it.

In the third chapter there are many examples of computer-based learning materials, arranged by discipline, an excellent guide for the person who wants to get started and needs to know where to begin. The Decision Guide offered in Appendix 4 will help him avoid the mistakes of others by answering some well formulated questions about the target situation into which the innovation will be introduced, the characteristics and purposes of the innovation, and the costs and support necessary to maintain it.

Chapter 4 presents a series of interesting case studies illustrating various approaches to the development of computer-based learning materials. Although the reader is left to draw his own conclusions the authors caution that "Whether you place your bets on creative individuals, discipline experts, systematic methodology, student or teacher involvement, powerful technology, multi-disciplinary teams, magnitude of funding, or a combination of these, *there are no guaranteed outcomes in quality or acceptance of the end product.* [emphasis added]"

Now for the bad news. In spite of this cautionary note and the statement that "it seems neither useful nor possible to seek principles or optimum strategies for development," the authors nevertheless conclude that the major problem in the development of quality materials justifying widespread adoption has been the "*lack of a coherent plan.*" They therefore advocate that a "coordinated national program" be established with "strategy coherence" based on one of five alternative models. All of the models involve massive infusions of Federal funds, curriculum by committee, and are extentions or combinations of existing approaches that certainly have neither proved themselves nor given any indication that "bigger is better."

Are there no other alternatives? Is there no place else to look for *authors* and distribution mechanisms than within the current structures which have proved so unsuccessful to date?

Perhaps looking at the past uses of computers in the classroom is not the best way to think about future uses of the computer in education. Perhaps a new viewpoint is necessary.

Let us consider for a moment some of the changes in the production of materials which accompanied another powerful technology as it moved from the hands of the few to the hands of the many, and recall some of the more notable changes that occurred when typography released the books from the hands of the scribe and the monastery into the hands of the public at large.[9] Until Gutenberg, there were no authors writing for a public. The public of Dante and Chaucer was necessarily a small group who listened to the poets reciting their verses, Reading publics, in our sense of the term, did not exist. The situation then was not unlike that of the composer before phonograph, radio and LP—the audience for new works was small and select.

The impact of having computer terminals located in homes, libraries, and places of public access, will not simply be to move the classroom to a new location but to change the audience and the conditions of use. This will happen regardless of anything that educators do or don't do. The use of pocket calculators by school children today is a case in point. It happened, and would have done so, with or without the approval of the National Council of Teachers of Mathematics.

What can we do then to assure the nationwide, beneficial use of computers in education? Let me offer a few suggestions of my own.

First, we must act now to gain the experience needed to make intelligent and considered decisions about computer use in education. When computer power becomes as freely and easily accessible as electric power is today, the question must not be "what can we afford to do," but "what do we want to do."

Investments in providing accessibility to computing power in our schools should not be viewed as an investment in machines, but rather as an investment in the experience to be gained by students, faculty and administrators. Long lead times are necessary to prepare for its use.

Second, we should take the opportunity to rethink everything that we do in education, remembering that the vast range of uses to which print has been put in education were not at all apparent on the day that Gutenberg set his first line of type. I would guess that we haven't even begun to scratch the surface in finding new and imaginative roles for the computer. We must look ahead rather than behind; we are at the start of an adventure, not the end.

Third, we must not consider the computer in isolation, but rather view it together with the other communications technologies which are shaping our lives—radio, broadcast television, cable television, home videodisc systems, telephone, satellite carriers and the rest.

And fourth, let us remember that computers alone will not bring about change. They need the efforts of imaginative people who can try out and demonstrate their ideas in viable settings, which are large enough to accurately portray the critical interactions which often fail to appear in small settings and whose absence creates misleading results.

The technology that will allow us to do whatever we want to do will soon be at hand. We must be prepared with the visions, the insights, and the experience to know what we want to do. *Learning Alternatives* serves a well-defined need. It tells us how to begin.

FOOTNOTES

1. National Academy of Sciences — National Research Council, *Digital Computer Needs in Universities and Colleges*, Washington, D.C., 1966.
2. President's Science Advisory Committee, *Computers in Higher Education*, The White House, Washington, D.C., 1967.
3. Oettinger, Anthony G. with Sema Marks, *Run, Computer, Run: The Mythology of Educational Innovation*, Harvard University Press, Cambridge, Mass., 1969.
4. Commission on Instructional Technology, *To Improve Learning: A Report to the President and the Congress of the United States*, U.S. Government Printing Office, 1970.
5. Levien, Roger E. *et. al. The Emerging Technology: Instructional Uses of the Computer in Higher Education*, McGraw-Hill Book Company, New York, 1972.
6. *The Fourth Revolution: Instructional Technology in Higher Education*, McGraw-Hill Book Company, 1972.
7. Levien, 1972, p. 3.
8. For those who consider the pocket calculator a toy, let me point out that many of the 13 million calculators now in use are equivalent in power to what John von Neumann had available to him in 1945 for the entire Manhattan Project.
9. The ideas here are stated by Marshall McLuhan with respect to the increased availability of 8mm film in the schools and appear in *8mm Sound Film and Education*, Louis Forsdale (ed.), Bureau of Publications, Teachers College, Columbia University, New York, 1962. Many of the comments on the value of accessibility as a key factor in the usefulness of an educational product were first expressed by Louis Forsdale in this volume.

CREATIVE COMPUTING Reviews

Reviews Editor: Peter Kugel, School of Management, Boston College, Chestnut Hill, MA 02167.

Readers: Want to be a reviewer? Write to the Reviews Editor directly. Publishers: send materials for review to the Reviews Editor.

Edcentric: A Journal of Educational Change, published by The Center for Educational Reform, Inc., P.O. Box 10085, Eugene, Oregon, 97401, six-issue (one-year) subscription $6.00 for individuals, $10.00 for institutions.

The name says it very succinctly: *Edcentric* is a journal of educational change and criticism, relevant and up-to-date because it is published and written by people in the schools, for people in the schools. The topics covered in the magazine range from, well, anything to everything of interest to people who believe that schools should treat people as human beings. The magazine is aimed at teachers and parents and concerned persons who are interested in "working to change education and to make changes through education" and sharing their experiences with others.

Edcentric is rather unusual in that it is put together by an editorial collective, the members of which are in a constant state of flux. As a consequence, the magazine comes out somewhat irregularly (it does come out—37 issues so far, the latest on Public Alternative Schools) but it has a certain life and spirit that marks *Edcentric* as a vital force for change and understanding.

John Lees
Rolla, MO

ANS COBOL. Ruth Ashley. 242 pp. $3.95. John Wiley & Sons, Inc. (Wiley Self-teaching guides), New York, 1974.

Ruth Ashley's American National Standard COBOL is an acceptable self-teaching book and classroom text for those who do not have access to a computer terminal and do have previous programming experience. A general understanding of programming and flowcharting is assumed. The author displays her knowledge of the elementary errors students have when beginning to program the most widely used computer language in business today.

Each chapter starts with a list of behavioral objectives and proceeds in the standard programmed text method of frames followed by multiple choice questions. The answers which directly follow the questions are supplemented with detailed and clear reasons for those answers. The student is provided with strips of COBOL coding forms for writing his or her programs. Throughout the book, much attention is given to spelling, punctuation, and format as is required in this language.

The author suggests that students try to complete each of the ten chapters in one or two study sessions of 1 to 3 hours each. The first chapter is devoted to the different divisions. The final question asks the student to write a complete small program. The second and third chapters cover MOVE, COMPUTE, GO TO, IF and arithmetic operations. The different types of files and editing are covered in the next four chapters. Chapter eight clearly presents PERFORM options. The last two chapters on tables and desk files complete a well planned text for anyone who is interested in an inexpensive text for high school or college students beginning COBOL. Teachers will find a complete lesson plan prepared by a teacher who understands student errors.

Paul A. Chase
Leicester, MASS.

[Available from the Creative Computing Book Service. Catalog No. 7H. Use order form on inside back cover.]

THE FUTURIST
The Promise of Fusion Power
How Cable TV May Change Our Lives
Nations That Go Crazy
The Fantastic Future of Architecture
What is Anticipatory Democracy? by Alvin Toffler
A Modest Utopia by Stuart Chase
Is Megalopolis Inevitable?
The Family As an Agent of Social Change
Herman Kahn— A Bright Future After All?
ALSO:
a journal of forecasts, trends and ideas about the future

YOU CAN DO MORE IN THE FUTURE WITH THE FUTURIST.

No one knows *exactly* what will happen tomorrow, but forward-looking experts from many professions are making a concerted effort to understand and forecast future social and technological developments.

Their forecasts and ideas now are easily and conveniently available in THE FUTURIST, an exciting bi-monthly magazine that specializes in keeping people informed about what may happen in the years ahead. By reading THE FUTURIST you will learn about forthcoming developments and be better prepared to deal successfully with our rapidly changing world.

You will find THE FUTURIST a source of valuable ideas for anticipating and adjusting to change, making long-range plans, and reaching effective decisions in your personal and professional life. Enter your subscription now, to get your own "window on tomorrow."

Recent articles from THE FUTURIST:

"Anticipatory Democracy" by Alvin Toffler

"Man-Woman Relationships in the Future" by Herbert Otto

"Architecture Beyond 2000" by Roy Mason

"The Road to World Harmony" by Jay Forrester

"Education for a Leisure Society" by Robert Strom

"Body Pleasure and the Origins of Violence" by Jim Prescott

"The Future of Computer Conferencing" by Murray Turoff

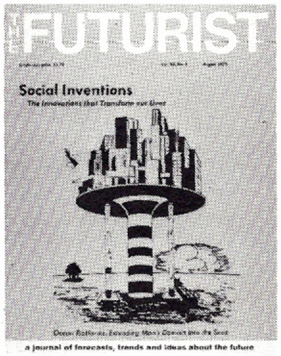
THE FUTURIST
Dept. L
4916 St. Elmo Avenue
Washington, D.C. 20014 • U.S.A.

Enter my subscription to THE FUTURIST for one year at only $12.

□ Payment enclosed. (I'll receive an extra issue as a bonus.)

□ Bill me.

GUARANTEE: I understand that if I'm not 100% satisfied with my first issue of THE FUTURIST, I'll let you know within one month and my payment will be refunded or my bill cancelled.



Name ______________________

Address ______________________

City __________ State ______ ZIP ______

Occupation ______________________

Sorting and Sort Systems, Harold Lorin. Addison-Wesley, 1975. 460 pp., $16.95.

This book is for experienced programmers who are interested in developing or using a sort. It would also be useful to non-programmers who need to understand the behavior of Sorting Systems, since it is one of the most readable texts on sorting in the literature with the details of programming left to the appendices. The specific sorting methods are covered broadly and carefully while recognizing that the ultimate performance of algorithms depends on the computer in use. Thus the reader comes away with a good intuitive feel for sorting and some confidence that, if there were not a good sort available in his subroutine library, he could knock off a respectable one in a few hours.

I feel that sorting should be one of the first things taught fledgling programmers, even in science-oriented schools or departments. Besides its practical value, it is an interesting mathematical subject. (No one yet knows how to make a SHELLSORT behave optimally. And who would expect that the fastest known procedure for reliably putting order into a sequence of values involves the generation of *random* numbers?)

The only criticism of the present book is that there are no exercises. In spite of that, the book ought to be available to any teacher of programming, if only because it is stimulating and pleasurable reading.

Not the least attractive feature of the book is the inclusion of all the ACM sorting algorithms (in ALGOL) as well as several related PL/I programs.

L. D. Yarbrough
Lexington, MA

*****

Mechanics. Herbert D. Peckham, Student Lab Book, 32pp, $1. Teacher's Advisor, 40pp, $1. Hewlett-Packard Co., Cupertino, Calif., 1972.

These booklets, in the HP Computer Curriculum Series, are intended to help meet the need for computer-oriented problems in physics by providing students an opportunity to use the computer as a problem solving tool within a particular subject matter area. Specifically, this unit is intended as an "enrichment" experience in the field of mechanics.

There are seven topics: rates, displacements, Newton's second law, half step method, the harmonic oscillator, more complicated forces, and orbital motion. In each section, there is a little preparatory explanation, an initial supplied program, and several exercises where the student is asked to modify the given program to extend or generalize the results. Each section also provides at least one advanced programming exercise where the student is asked to construct a separate but related program from scratch. The Teacher's Advisor provides program listings and sample runs. With only minor adjustments, these programs should be adaptable to different computers.

The author assumes that the student (1) has had algebra and some trigonometry, (2) that he already knows how to write a simple computer program in BASIC, and (3) that he will have access to a computer for at least two hours per week.

Perhaps the most thorny problem in teaching introductory physics is that the student does not have the requisite tools of calculus at his disposal. This booklet serves as an example of how the computer can help overcome the difficulty. Although the words "calculus," "derivative," and "integration" are never used, nearly all of the exercises here involve calculus by means of difference approximations. The student who uses this material will accordingly have a much better feeling for calculus when he or she studies it analytically.

The computer enriches the study in other ways. Using just one program, the student can perform many different experiments by making slight modifications to the basic program, and make generalizations on the basis of the results.

The exercises are sequenced so that the student applies what he or she has learned in the previous problem to solving the next one. This feature provides a strong thread of continuity and makes the treatment highly attractive. On the other hand, because it does build on previous results and is cumulative, one cannot omit any section and pick up a later section.

Although the exercises, in a "watch me and then imitate, modify, and extend" format begin slowly, they quickly become challenging in terms of programming skills required. The first two sections on rates and displacements involve concepts and problems that are simple and straightforward. In section 3, on Newton's second law, the programming begins to get a little more involved. Perhaps the author discovered this, for he has added an appendix where he looks at the section 3 initial program in more detail. There is a flow chart and a line-by-line description of what is happening in the program. The reviewer agrees that this would be a good time to nail down a few key programming skills by looking at a fairly central program. The time would be well spent since subsequent exercises are somewhat more involved. An average high school physics class in my area would find these later exercises moderately difficult. Indeed, the author assumes that "students taking "introductory physics (in college?) will be quite capable as a group."

This unit could be used with a uniformly good high school physics class, but as the instructor, I would want to provide some additional instruction on the concepts as well as an occasional review of programming principles and techniques. It would also be useful as a directed individual study project for a bright high school senior with some programming experience and an available terminal.

With regard to the pamphlet's printed format, it would be helpful if at least the statements of the exercises could be reprinted in the Teacher's Advisor so that the teacher could relate the solution commentary to the problems, and not have to refer back and forth between manuals.

Daniel S. Yates
Glen Allen, VA

*****

User's Guide to Computer Crime, Its Commission, Detection and Prevention. Stephen W. Leibholz and Louis D. Wilson, Chilton Book Company, Radnor, Pa. 1974. $9.95

This book comes complete with disclaimer that all the cases of theft, fraud, sabotage, and espionage conducted via computer as described in the book have actually been comitted *and discovered*. However, the implication to the casual reader is clear: There must be more clever people who have not yet been discovered!

Were it not for the fact that the technology is mercifully somewhat sketchy and general in places, the *User's Guide to Computer Crime* could be labelled as hazardous to the health of computers. As it stands, it could be a generalized cookbook for how to make money, or at least mischief, by manipulating the computer.

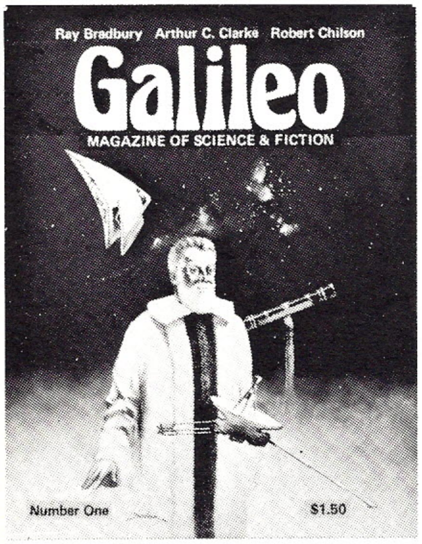
Those who are already deeply involved in computer security will find nothing new here. But those who had labored under the mistaken notion that because the computer is complex, it is also safe, can profit from it. The incidents of unauthorized computer manipulation make some of our flights of fancy seem positively earthbound. The book attempts to move the reader who may be responsible for a data processing system from the edge of immediate panic to a more positive predisposition to do something, perhaps using some of the valuable suggestions contained in the latter portions of the book.

The book has four major divisions: basic problems and case histories of various "crimes," specific measures and general operating principles to achieve reasonable protection from incidents, methods of auditing and detecting unusual activity along with some legal discussion of computer use and abuse, and recommendations for user programs of action in accounting and legislative realms. The appendices are as revelatory as the book itself: A checklist for ensuring data processing operations security, a reprint of the Fair Credit Reporting Act, and even the reprint of the first of 105 counts in the Equity Funding indictment, one of the largest computer-assisted heists in history (so far), and the Stanford Research Institute survey of programmer ethics. In the latter, of the 55 managers and programmers surveyed, 13 thought it was "okay" to attempt to log onto a time-sharing service for which they were not authorized users. 10 had actually done it. Only 17 thought it was illegal.

Not only is the book worthwhile for anyone who has had cause to manage, use or mistrust computers, but it is written in an easy, readable style.

Deanna J. Dragunas
Wetumpka, AL

THE BEST OF THE FUTURE AVAILABLE NOW...


Ray Bradbury Arthur C. Clarke Robert Chilson
Galileo
MAGAZINE OF SCIENCE & FICTION
Number One
$1.50

There is a new direction in the field of Science Fiction. The early period, the time of Gernsback, Burroughs, and robots, gave way to the middle period of interstellar travel, Russell, and Campbell. Now the new wave of authors, led by the most farsighted of the 'old timers', are taking Science Fiction into its young adulthood. Experimentation is now mixed creatively with the tested craftsmanship of the masters. The great work of the past is now the foundation of the future. The best is yet to come, and GALILEO is where you'll find it.

GALILEO is a new science fiction quarterly (destined to be monthly) which will be available only through subscription and a few selected bookshops around the country. Its cost, at $1.50, will reflect the quality of its contents in design, illustration, and most importantly, writing. (Subscribers will, of course, benefit from a reduced cost per copy). The great names we proudly announce on our covers are symbols of the fine works by authors both new and established which we will gather together for each voyage to the stars. An outstanding crew in every issue.

Our intention in publishing GALILEO is to pick up where editors like John W. Campbell left off. Science Fiction is—or should be—quality entertainment AND literature. It is the most important literature of our time, or any time. . . . History teaches, science leads. Our GALILEO offers a vision of mankind's future through the eyes of the poet and the scientist. There can be no true appreciation of the present, or the past, without the perspective of the future.

Science and theory are the raw materials, . .what mankind might do with them is Science Fiction. GALILEO is the quality Science Fiction magazine, the best of the future, available now!

BRADBURY CLARKE CLEMENT & MORE!

This is one voyage I don't want to miss. Sign me up for the special subscription rate I have checked below. My check is enclosed.

[ ] 4 issues at $4. (saving $2.) [ ] 10 issues $9. (saving $6.)

Name:____________________

Street:__________ Town:__________

State:__________ Zip:__________

Send to: GALILEO, 339 Newbury Street, Boston, MA 02115

First issue premiére
September 2, 1976

Security, Accuracy, and Privacy in Computer Systems. James Martin, Prentice-Hall, Englewood Cliffs, N.J. 1973, $19.95

The savvy professional who wants considerable, sound technical detail computer security would do well to invest in James Martin's *Security, Accuracy, and Privacy in Computer Systems.* This book can function either as a reasonably complete course in computer security, or as a reference book on specific areas. The 600-page length of the book makes reading it more than a casual task, but the writing is well done and never gets in the way of the information.

Martin divides his tome into five well-balanced parts, each containing information applicable to a range of system types, from batch processing to teleprocessing in real-time. These major sections are:

I. Definition of the overall problem
II. Design of the computer system
III. Design of physical security
IV. Design of administrative controls
V. Design of the legal and social environment

The book also contains extensive appendices giving guidelines for the construction of storage vaults, a checklist which relates security vulnerabilities to specific sections of the book, and copies of legislation pertaining to privacy. These are not currently relevant because the Privacy Act of 1974 (H.R. 93-579) was passed after this book was issued, and because Congress has some firm proposals for legislation to apply to the private protection in computer systems.

James Martin is well-known for his expertise and prolific writing in the area of data communications which makes his advice and analysis particularly relevant to today's many remotely accessed computer systems. Too often in the past, data system protection has meant fire and flood protection. These traditional security concerns, as well as those relating to personnel management, locks, and print-out disposal are given their due. But concerns for data scrambling, communication line protection, operating system "glitches," and other areas which are more intimately involved in the technology of modern data processing systems are emphasized.

Deanna J. Dragunas
Wetumpka, AL

*****

Scelbi's Galaxy Game by Robert Findley, 1976, 172 pp. paper, $14.95. Scelbi Computer Consulting, 1322 Rear Boston Post Road, Milford, CT 06460.

Scelbi's First Book of Computer Games by Nat Wadsworth and Robert Findley, 1976, 122 pp. paper, $14.95. Scelbi Computer Consulting.

If you own a small 8008 or 8080 based system, you'll be interested in these two new computer game books by Scelbi. The games in these books aren't in Basic or any other high-level language, but instead are in assembly language and machine code. Extensive documentation is included.

The object of Galaxy is to travel throughout the galaxy in search of alien ships and destroy them in a limited number of stardates. You have two types of weapons: torpedoes, and a "phasor" (fired by the Starship Enterprise we assume). The game is pretty bony—no floating point math, very short messages, and some things which will bother Star Trek—excuse me, Galaxy purists, but remember, it all fits in 4K!

The other book, *Scelbi's First Book*, contains Space Capture, Hexpawn, and Hangman. In Space Capture, played on an 8 by 8 grid, you attempt to prevent an enemy spacecraft from moving by destroying all the sectors around him with your "phasor." Hexpawn appears to be very much like the Hexpawn game described by R.R. Wier in *Byte*, while Hangman is modeled on the popular word game.

Although the messages contained in these games are brief and a little too cute, (example: !#0# DARN! YOU HAVE ME CAPTURED!!) you can change them easily. Since each program is thoroughly explained, it might be interesting to add some of your own features if you have enough memory.

Steve North
Newfoundland, NJ

Recursive Programming Techniques, W. H. Burge. Addison-Wesley Publ. Co., 1975. 272 pp., $15.75

This is a graduate-level book on programming which will appeal primarily to Computer Science majors. The presentation is taut, rigorous, and thorough, and requires a high level of motivation from the reader. As in most of the books in this (The Systems Programming) Series, there are no exercises, so it is more of a reference text than teaching text, but it is excellent in either role.

Some excerpts from the table of contents may give the flavor of the book:

1. Basic Notions and Notations
   1.3 Variables and Lambda Expressions
   1.4 Data Structures
   1.10 Recursive Functions
2. Program Structure
   2.2 Reverse Polish Programs
   2.8 Compiling Expressions
   2.10 Labels and GO TO Statements
3. Data Structures
   3.5 List Structures
   3.6 Trees and Forests
   3.10 Sequences, Coroutines, and Streams
4. Parsing
   4.3 Context-Free Languages
   4.6 Left-Corner Bottom-Up Parsing
5. Sorting
   5.3 Binary Search Trees
   5.5 Quicksort
   5.9 Tape Sorting

L. D. Yarbrough
Lexington, MA

*****

Introduction to Data Processing. Martin L. Harris, 326 pp. $3.95. John Wiley and Sons, Inc., New York. 1973.

Harris presents the fundamentals of data processing in a programmed format. Each chapter is divided into numbered sections that present new information. The objectives are identified at the beginning of each chapter and each chapter also contains a review. You may use the review as a pretest and, if satisfied with your performance, you may skip that chapter.

The basic concepts of data processing are presented in an elementary, low-key fashion. The book could truly serve as an "introduction" for the neophyte. Little previous knowledge of data processing or computer application is assumed. "The purpose of this book, then, is to give you some basic understanding of what data processing is, how it is organized, what types of equipment are used, and how a particular data processing system is designed." A cross referencing chart to other data processing texts is provided for the reader who seeks additional information or another perspective to the same topics.

One chapter provides instruction in the BASIC programming language as an illustration of how instructions are written for a computer. Uses for other languages (COBOL, FORTRAN, RPG) are only briefly mentioned.

The text can be recommended to someone who is beginning a study of data processing. It provides an introductory background in a programmed format for self-study. It might also be used as a review text. It is unfortunate, however, that only masculine pronouns (he, his, him) are used throughout the book.

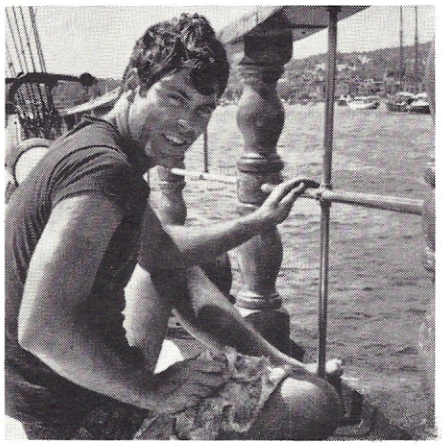
Jane Donnelly Gawronski
San Diego Department of Education
San Diego, CA. 92111

For once in your life...live.

A sleek graceful sailing vessel glides across the sometimes green, sometimes blue Caribbean. The cargo: you. And an intimate group of lively, fun-loving shipmates.

Uniform of the day: Shorts and tee shirts. Or your bikini if you want. And bare feet.

Mission: A leisurely cruise to remote islands with names like Martinique, Grenada, Antigua—those are the ones you've heard of. Before the cruise ends, you'll know the names of many more. You'll know intimitely the enchanting different mood of each...and its own beauty and charm.



Life aboard your big sailing yacht is informal Relaxed. Romantic.

There's good food. And 'grog'. And a few pleasant comforts... but any resemblance to a plush pretentious resort hotel is accidental.

Spend 6 days exploring paradise. Spend six nights watching the moon rise and getting to know interesting people. It could be the most meaningful experience of your life ...and it's easily the best vacation you've had.

A cruise is forming now. Your share from $265. Write Cap'n Mike for your free adventure booklet in full color.

Come on and live.

Windjammer Cruises.

Name ______________________________

Address ______________________________

City ________________ State ____________ Zip ____________

Phone ______________________________

P.O. Box 120, Dept. 17 Miami Beach, Florida 33139

Electronic Computers. S. H. Hollingdale and G. C. Tootill. Penguin Books (A Pelican Original), Middlesex, England. Published 1965 and revised in 1975. 378 pages, illustrated, indexed, 4½ x 7, £1 ($3.95 in the United States and Canada), paperback.

The publisher's notes on the back cover describe this book as being intended for "the general reader." "[The authors] have taken particular care with the specialist jargon of their subject, explaining each term as it occurs."

A "general reader" had better equip himself or herself with a solid background in mathematics and electronics if that general reader wants to wade through some parts of the book. A certain technical patois in a work of this sort is probably unavoidable, but it is irresponsible if not actually misleading of the publisher to represent the book as a general work when it is in fact no such thing.

For example, it is hardly helpful to offer, as the authors do in a footnote on page 107, a twenty-five word definition of a derivative. A reader who doesn't already know what a derivative is will not understand either the definition or the second order differential equations sprinkled across the page in the solution of a multi-storied building stress analysis problem. And no reader who doesn't know *something* about electronics is likely to understand the wiring diagram on the next page of the analog computer which is supposed to work out the solution.

The authors are both mathematicians who have been involved in the British computer industry almost from its beginnings after World War II. The writing is clear, although a bit leaden in the inimitable style of British technical writing. The book's emphasis is on the British computer industry and British computer users, with little said of what is happening on the other shore of the Atlantic. The bias is somewhat strange in light of the almost total American domination of the field.

There are twelve chapters. The first two present a detailed history of computing, beginning with the abacus. It has always been a mystery to me why the decimal notation was not used in ancient times by the same peoples who used the abacus so skillfully. Nothing seems to be more natural than the transition from the positional decimal notation of the abacus to a similar system for writing the same numbers on paper or papyrus or cuneiform. Yet the ancient Greeks, who were first rate mathematical heavyweights, never made the connection and used a cumbersome alphabet-based system for transcribing numbers. Hollingdale and Tootill offer the explanation that since arithmetic was always done on an abacus and never on "scratch papyrus," the awkwardness of the method used to write down numbers was never really a problem. I find this explanation clever but not quite satisfying.

In fact the Babylonians did use a positional notation with a radix of sixty but without a zero, so 1, 60, 3600 and 1/60 were all written the same way and the correct figure had to be deduced from the context. The elusive zero wasn't invented until many centuries later by the Hindus, even though it was right there on the abacus all the time.

Of course we can always ask questions of the type — If the Greeks were really so smart, why didn't they invent the cheeseburger?" In retrospect many brilliant ideas seem to be so simple that it is difficult to understand why no one thought of them before. Hindsight is one of the most exact of the sciences.

In 1946 the American Army in Japan staged a competition between a Japanese version of the abacus, known as a soroban and the most modern electric (not electronic — the word wasn't even in the dictionary then) calculator. To quote *Stars and Stripes,* "the abacus victory was complete." So ended an era.

The emphasis in the book is on hardware. There are numerous diagrams of the "insides" of various units, including a differential analyzer! Not much space is devoted to minicomputers, and the little there is on the subject of microminiaturization seems to have been added as an afterthought when the book was last revised in 1975. This last comment is not offered as a criticism, since no one in his right mind could have foreseen the Intel 8080, but the sparse treatment does date the book.

In fact the 1975 revision appears not to have been especially extensive. In several instances the authors mention the technique of punching out intermediate results on cards for use in later steps of the calculation, as though tapes and disks had not yet been invented. Nevertheless, the workings of tape units are explained in chapter seven and almost the whole of chapter twelve is devoted to a very detailed description of disk drives.

Although there is a chapter on software, it is mostly devoted to a description of programming languages, especially ALGOL. There is too little said about operating systems or time sharing, and nothing at all of interpreters or of structured programming.

The effect of computers on society is not treated at all, except in an aside in which the authors mention "effective ways of using [computers] — for good or ill."

To sum up, the book is clearly written and informative, although hardly as up to date or inclusive as it might be. It is also too technical for the general reader, but it can be useful to the reader with a good background in computers who wants to fill in some gaps in his or her knowledge, particularly about early hardware.

A final point — if one British pound is worth* about two American dollars, why does this book cost £1 in Britain and $3.95 in the United States?

Alex Ragen
Jerusalem, Israel

* *When this review was written, matters changed rather seriously soon after.*

*****

Configurations (Game). Harold L. Dorwart. Wiff 'N' Proof Learning Games Association, 1490 South Boulevard, Ann Arbor, MI 48104. $6.75.

After playing the game *Configurations* a student should have: 1) The realization of the existence of finite projective geometries, 2) An aroused interest in finite projective geometries, 3) A better understanding of some number patterns.

Configurations contains a series of geometric puzzles, as games of solitaire and discovery, based on finite projective geometries. The geometries used are the Fano 7_3, the Mobius-Kantor 8_3, the Pappus 9_3, the Desargues 10_3 and others. Each of the games is played on a game board with small plastic numerals provided in the game kit. By placing the numerals on the boards according to the rules found in the instruction manual, the player is led to very interesting mathematical discoveries.

Configurations would be a worthwhile addition to any Math Lab or Math Resource Center to be used by students working on independent projects or by those who are interested in a fun way to acquire new knowledge.

Peter B. Danos
W. Redding, Conn.

*****

Software Tools, Kernighan, B.W. and Plauger, P.J., Addison-Wesley Publ. Co., 1976, $8.95.

To teachers of programming: Stop what you are now doing; get yourself a copy of this book and study it. Then give some thought to what you have been teaching and what you should be teaching about programming. You will find in the book a lot of good ideas and a wealth of well-written, useful examples that you can use to explain to your students how to write clear, effective, reliable, understandable programs. If you get caught up by the ideas in the book you can send $25 to Ms. Linda Banks at Addison-Wesley and she will send you a 9-track 800 BPI magnetic tape containing all the programs in this book (over 200 in all) for you and your students to use and study.

To students of programming: If your teacher is too slow in responding to the above, get your own copy of this book and study it. Shortly you will be programming rings around your classmates and your teacher, and you will be preparing yourself to take your place among a new wave of professional computer scientists who know, for a change, what they are doing.

To professional programmers: Get on the ball; the new wave is coming.

To the authors and publishers of software tools: I am adding your book to a short list of things I wish I had done, along with the pocket calculator, the cassette tape recorder, and a few other goodies. This is the best programming book published in the last 3 years, at least. Thank you.

L. D. Yarbrough
Lexington, MA

CREATIVE COMPUTING BRINGS TOGETHER LEADING PROJECTS AND WRITERS UNDER ONE COVER

The following are just a small sample of the contributors that have recently appeared on the pages of *Creative Computing:*

- **David H. Ahl**
  Author of *101 BASIC Computer Games* and other books
- **Isaac Asimov**
  Author of over 165 books on science and science fiction
- **Alfred Bork**
  Director of Physics Computer Development Project, UC, Irvine
- **T. C. Chen**
  IBM San Jose Research Laboratory
- **Gordon R. Dickson**
  Noted science fiction author
- **Herbert L. Dreyfus**
  Artificial Intelligence Laboratory, UC, Berkeley
- **Thomas A. Dwyer**
  Project Solo and Soloworks, Univ. of Pittsburgh
- **Stephen B. Gray**
  Founder, Amateur Computer Society
- **Ron Jones**
  Founder, Zephyros Education Exchange and DeSchool Primers
- **Kenneth Knowlton**
  Computer artist, inventor of EXPLOR graphics language
- **Walter Koetke**
  Pioneer in using computers in secondary school mathematics
- **Peter Kugel**
  Inventor of many logic games
- **Ruth Leavitt**
  Computer artist, author of *Artist and Computer*
- **Sol Libes**
  Author of many electronics texts, President NJ Amateur Computer Group
- **Sema Marks**
  Co-author of *Run Computer, Run* and numerous articles
- **John McCarthy**
  Artifical Intelligence Laboratory, Stanford Univ.
- **Robert McLean**
  Ontario Institute for Studies in Education
- **Erik McWilliams**
  TIE Branch, National Science Foundation
- **Monty Newborn**
  Author of *Computer Chess* and other related books
- **Frederik Pohl**
  Noted science fiction author and editor
- **Bertram Raphael**
  Author of *The Thinking Computer*
- **Robert Taylor**
  Director of Academic Computing, Teachers College, Columbia Univ.
- **John Tunney**
  U.S. Senator from California
- **Murray Turoff**
  Developer of computer conferencing
- **Xerox Palo Alto Research Center**
  Home of Dynabook and the Smalltalk language
- **John Whitney**
  Computer film-maker

PRAGMATIC, SIGNIFICANT REVIEWS EVERY ISSUE

Creative Computing brings you no-nonsense reviews of hardware, books, and games like the following:

- Microprocessor kits: Altair 8800, SWTP 6800, etc.
- Programmable hand calculators: HP 22, TI SR 52, etc.
- Programmable desk calculators: HP 9815A, etc.
- Self-contained minicomputers: Tektronix 4015, Wang 2200, etc.
- Peripherals: Teletype KSR 43, TV Dazzler, etc.
- Video games: Pong, Odyssey 300, etc.
- Books: 10 reviews every issue; every BASIC book reviewed in 1976
- Miscellaneous: Movies, TV shows, games, records, etc.

CREATIVE COMPUTING IN A NUTSHELL

Creative Computing is for students, educators, hobbyists, and anyone curious about computers. Carries articles and stories about computers in art, music, education, space exploration, health care, and other unusual as well as traditional applications. Lots of how-to building and programming information. Contains programs, games and simulations with complete listings mostly in BASIC, FORTRAN, and APL. In-depth book reviews, problems, puzzles, fiction, poetry, side-splitting cartoons and more. Contemporary, humanistic editorial approach. Published bi-monthly. At least 96 pages per issue. Over 25,000 circulation in 50 states and 22 foreign countries.

NO RISK GUARANTEE

"You may cancel your subscription at any time and we will refund the balance without question."

ORDER FORM

MAGAZINE SUBSCRIPTIONS

Creative Computing (Bi-monthly)

□ Institutional	Any Country	1-year	$15
□ Institutional		3-year	40
□ Individual	USA only	1-year	8
□ Individual		3-year	21
□ Individual	Foreign	1-year	10
□ Individual		3-year	27

Creative Computing and **Byte** (Save 10%)

USA	□ 1-year	$18	□ 3-year	$46
Foreign	□ 1-year	23	□ 3-year	61

BOOKS AND MERCHANDISE

Quan.	*Cat.*	*Descriptions*	*Price*

Books shipping charge $1.00 USA, $2.00 Foreign ________

NJ Residents add 5% sales tax ________

TOTAL (magazines and books) ________

□ Cash, check, or M.O. enclosed

□ BankAmericard } Card No. ________

□ Master Charge } Expiration date ________

□ Please bill me ($1.00 billing fee will be added)

Name ________

Address ________

City ________ State ________ Zip ________

M6800 Microprocessor Programming Manual
SWTPC 6800 COMPUTER SYSTEM
SWTPC 6800 COMPUTER SYSTEM
RESET
POWER

SWTPC 6800

- ★ COMPLETE WITH 2K OF MEMORY
- ★ SERIAL INTERFACE
- ★ STANDARD ROM MONITOR (Motorola MC 6830L7)
- ★ 4K AND 8K BASIC AVAILABLE

Always the best value in hardware and now an outstanding selection of software too. What more could you want in a computer system? For less than four hundred dollars you get everything you need—ready to connect to a terminal and go to work. No surprises no funny business, just good reliable hardware in a very practical system that may be expanded to meet almost any later need.

Memory and interfaces are not extra cost items in our system. A standard Motorola MIKBUG® ROM monitor makes the system completely compatible with Motorola software and eliminates any need for console switches and light. Data may be entered from the terminal in convenient hexidecimal form. The power supply is adequate to operate a fully expanded system with up to 24K of memory and up to eight (8) interfaces—simultaneously.

See the 6800 and our peripheral equipment at your nearest dealer, or write for a complete description.

MP-68 COMPUTER KIT—with serial interface, 2k of memory and ROM monitor $395.00 ppd

® Motorola

Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, Texas 78216